

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

Logiciels libres et droit d'auteur

Laurent, Philippe

Published in:

Les logiciels libres face au droit

Publication date:

2005

Document Version

le PDF de l'éditeur

[Link to publication](#)

Citation for pulished version (HARVARD):

Laurent, P 2005, Logiciels libres et droit d'auteur: naissance, titularité et exercice des droits patrimoniaux. Dans *Les logiciels libres face au droit*. Cahiers du CRID, Numéro 25, Académia Bruylant, Bruxelles, p. 23-89.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

PARTIE II

LOGICIELS LIBRES ET DROIT D'AUTEUR : NAISSANCE, TITULARITÉ ET EXERCICE DES DROITS PATRIMONIAUX

Philippe LAURENT

INTRODUCTION

36. Un logiciel libre est avant tout un programme d'ordinateur ou « logiciel », c'est-à-dire un ensemble organisé d'instructions qui seront lues par un ordinateur et lui feront accomplir certaines tâches. De même que le texte constitue la substance d'un livre, le code constitue la substance d'un programme d'ordinateur. Cependant, une distinction essentielle doit être établie d'emblée entre plusieurs types de codes. L'homme travaille, comprend et programme en code source, par le biais de certains langages de programmation. A l'inverse, la machine ne lit que du code objet et n'exécute que du code binaire, incompréhensibles pour les programmeurs. Entre le code source, le code objet et le code exécutable intervient la compilation, c'est-à-dire la traduction du code source en code machine.

37. Un logiciel libre, aussi bien sous forme de code source que de code objet ou exécutable, est donc un « programme d'ordinateur » protégé en tant que tel par le droit d'auteur. En 1991, le législateur communautaire consacra le principe selon lequel les programmes d'ordinateur devaient être protégés par le droit d'auteur en tant qu'œuvres littéraires au sens de la convention de Berne pour la protection des œuvres littéraires et artistiques⁴². Les traités internationaux ont suivi le pas : l'article 10, alinéa 1, de l'accord ADPIC (TRIPS) du 15 avril 1994 et l'article 4 du traité de l'OMPI (WCT) du 20 décembre 1996 imposent également que les programmes informatiques soient traités « en tant qu'œuvres littéraires ».

⁴² Article 1, §1, de la directive 91/250/CEE du Conseil du 14 mai 1991 concernant la protection juridique des programmes d'ordinateur, *J.O.C.E.*, 15 mai 1991, n° L 122/42, ci-dessous dénommée « la Directive ». Convention de Berne pour la protection des œuvres littéraires et artistiques du 9 septembre 1886, dont la dernière version faite à Paris le 24 juillet 1971 a été introduite en droit interne par la loi du 25 mars 1999, *M.B.*, 10 novembre 1999, p. 41891 et s.

Dès lors tout programmeur bénéficie des prérogatives du droit d'auteur sur son œuvre informatique : naissance des droits au moment de la création, sans aucune formalité, pourvu que l'œuvre soit originale et mise en forme, durée de protection de 70 ans à partir du décès de l'auteur, jouissance de droits patrimoniaux (droit de reproduction⁴³, de distribution⁴⁴ et de communication au public⁴⁵) et de droits moraux.

38. S'il présente des similitudes avec les œuvres littéraires, dans la mesure où il est « écrit » dans un certain langage selon des règles de composition bien déterminées, le programme d'ordinateur présente cependant la particularité d'être fonctionnel⁴⁶ : sa raison d'être reposera toujours sur ses fonctionnalités, c'est-à-dire son aptitude à effectuer certaines opérations. L'on apprécie donc moins un logiciel pour ses qualités artistiques, esthétiques ou littéraires, que pour son efficacité, les résultats obtenus et, au final, sa capacité à remplir la tâche pour laquelle il a été conçu. Cette caractéristique a conduit les législateurs à prévoir en droit d'auteur une série de dispositions propres aux programmes d'ordinateur. Entre autres, le régime des exceptions, des droits moraux et des créations d'employés a été adapté.

En Belgique, le législateur a séparé les dispositions de droit d'auteur spécifiques aux programmes d'ordinateur en leur consacrant une loi spéciale (ci-après « LPO »)⁴⁷, au lieu de les insérer dans la loi générale (ci-après « LDA »)⁴⁸, consacrant par ce fait même la spécificité des œuvres informatiques. La LPO n'est cependant pas complète et ne consacre pas un système de droit d'auteur autonome : il s'agira donc d'appliquer en priorité les principes édictés par la loi spéciale et, quand celle-ci reste muette, de se référer aux dispositions de la loi générale⁴⁹.

⁴³ Article 4, a) et b) de la Directive.

⁴⁴ Article 4, c) de la Directive.

⁴⁵ En application de la loi générale sur le droit d'auteur (loi du 30 juin 1994 relative au droit d'auteur et aux droits voisins, *M.B.*, 27 juillet 1994, p. 19297 et s.).

Voir par exemple F. DE VISSCHER et B. MICHAUX, *Précis du droit d'auteur et des droits voisins*, Bruxelles, Bruylant, 2000, p. 222 ; F. BRISON et J.-P. TRIAILLE, « La nouvelle loi sur la protection des programmes d'ordinateur, dans le sillage de la loi sur le droit d'auteur », *J.T.*, 1995, p. 143.

⁴⁶ A. STROWEL, et E. DERCLAYE, *Droit d'auteur et numérique : logiciels, bases de données, multimédia*, Bruxelles, Bruylant, 2001, p. 36.

⁴⁷ Loi du 30 juin 1994 transposant en droit belge la directive européenne du 14 mai 1991 concernant la protection juridique des programmes d'ordinateurs, *M.B.*, 27 juillet 1994, p. 19315 et s. (LPO ou « loi spéciale »).

⁴⁸ Loi du 30 juin 1994 relative au droit d'auteur et aux droits voisins, *M.B.*, 27 juillet 1994, p. 19297 et s. (LDA ou « loi générale »).

⁴⁹ Voir à cet égard F. BRISON et J.-P. TRIAILLE, *op. cit.*, p. 141 et s.

39. La présente contribution a pour premier but de présenter la théorie générale relative à la naissance et la titularité des droits d'auteur sur un logiciel, et d'appliquer ces principes aux spécificités des logiciels libres. Nous y consacrerons un premier chapitre.

Un deuxième chapitre présentera les différents droits patrimoniaux dont jouit tout titulaire de droits sur une œuvre informatique, les exceptions à ces droits et la façon la plus courante d'exercer ces droits, à savoir, la concession de licences.

A la lumière des principes que nous aurons exposés, et des conclusions que nous aurons tirées dans les deux premiers chapitres, nous aborderons enfin, dans un troisième chapitre, la question de la viralité telle que généralement conçue dans le milieu libre, ainsi que certains cas d'espèce que nous pensons pouvoir y assimiler.

CHAPITRE 1 : LA NAISSANCE DES DROITS D'AUTEUR AFFÉRENTS AUX LOGICIELS LIBRES ET LEUR TITULARITÉ

40. Dans cette première partie, nous analyserons les conditions d'existence des droits d'auteur (section 1) ainsi que les règles de détermination des personnes qui en sont investies (section 2).

Section 1 : Naissance

41. Au sein de la grande famille des droits intellectuels, le droit d'auteur a pour principale caractéristique le fait que sa protection est accordée de façon automatique et sans formalité. Les droits d'auteur naissent dans le chef de l'auteur par le fait même de la création, c'est-à-dire par la mise en forme de l'œuvre originale. Aucun dépôt, enregistrement, entiercement ou autre acte officiel n'est nécessaire afin d'obtenir la protection qui est, dès lors, acquise gratuitement. Même si ces actes peuvent souvent constituer de bons moyens de s'assurer des preuves de la naissance et de la titularité des droits, leur pertinence ne dépasse pas l'ordre de la preuve.

Dès lors, les deux conditions nécessaires et suffisantes devant être remplies afin qu'un programme informatique soit protégé par le droit d'auteur sont l'originalité et la mise en forme.

Cette absence de formalité et la gratuité de protection qui en découle peuvent être considérées comme l'un des piliers nécessaires au maintien du Libre. Le

modèle économique du logiciel libre n'étant pas basé sur l'octroi de licences payantes relatives aux droits d'auteur, et le développement de ces logiciels reposant sur la collaboration de tous et la création d'œuvres multiples et en série, il eut été impossible de concevoir pareil concept sans le bénéfice d'un système de protection souple, systématique et gratuit.

§ 1. *Mise en forme*

42. La condition de mise en forme impose que le programme dépasse le stade de la pure conception intellectuelle : il doit faire l'objet d'une mise en forme, c'est-à-dire être exprimé de manière quelconque⁵⁰. Le code source, objet ou exécutable constitue bien entendu une forme d'expression correspondant à cette définition.

Cependant, cette exigence n'implique pas que la protection se limite au seul code. En effet, l'article 2 de la LPO souligne que la protection accordée par cette loi s'applique à toute forme d'expression d'un programme d'ordinateur.

De même, l'article 1 de la LPO élargit la protection accordée au matériel de conception préparatoire⁵¹. Par exemple, les documents d'élaboration sur support papier ou numérique, tels que les diagrammes de flux (flowcharts), les organigrammes, la documentation de conception, les dossiers d'analyse fonctionnelle et organique ainsi que tout autre élément « non-littéral »⁵² feront également l'objet de la protection⁵³. De même, les programmes pré-montés, maquettes ou prototypes seront protégés. Le logiciel ne doit donc pas être « opérationnel », mais seulement « suffisamment concrétisé » pour bénéficier de la protection par le droit d'auteur⁵⁴.

⁵⁰ Voir par exemple F. DEVISSCHER et B. MICHAUX, *op. cit.*, p. 8 et s.

⁵¹ La LPO devant s'interpréter conformément à la Directive, il faut noter qu'alors que l'article 1^{er}, al. 1, de cette dernière n'apporte pas plus de précisions quant à la notion, son 7^{ème} considérant vise « les travaux préparatoires de conception aboutissant au développement d'un programme, à condition qu'ils soient de nature à permettre la réalisation d'un programme d'ordinateur à un stade ultérieur ».

⁵² Les éléments non-littéraux sont les éléments qui ne sont pas exprimés en langage informatique. Voir A. STROWEL et E. DERCLAYE, *Droit d'auteur et numérique*, Bruxelles, Bruylant, 2001, p. 181. Pour un exemple de jurisprudence acceptant la protection des diagrammes de flux par le droit d'auteur, voir Civ. Bruxelles (cess.), 30 juin 2003, *I.R.D.I.*, 2003, p. 177 et s.

⁵³ A. BENSOUSSAN, *Le logiciel et le droit*, 2^{ème} édition, Paris, Hermès, 1998, p.37 ; voir également Corr. Hasselt, 16 février 1999, *I.R.D.I.*, 1999, p. 34 ; R.W., 1998-99, p. 1424.

⁵⁴ T.G.I. Paris, 31 mai 1995, *Expertises*, 1995, p. 319 ; A. LUCAS, et H.-J. LUCAS, *Traité de la propriété littéraire et artistique*, Paris, Litec, 2^{ème} Edition, 2001, p. 66.

Précisons que cette exigence de mise en forme découle de la notion même de ce qui fait l'objet de la protection par le droit d'auteur. En effet, c'est précisément cette forme concrète qui sera protégée par le droit d'auteur, et non les idées ou principes qui sont à la base du programme d'ordinateur ou de ses interfaces. Cette exclusion des idées et principes est explicitement mentionnée à l'article 2, alinéa 2 de la LPO⁵⁵.

Le fait que le logiciel en question soit un logiciel libre ne change en rien l'application de ce critère de mise en forme.

§ 2. *Originalité*

43. Lorsque le programme est mis en forme, encore faut-il qu'il soit original⁵⁶ pour bénéficier de la protection par le droit d'auteur.

44. Notons d'emblée qu'un logiciel ne doit pas être abordé dans sa globalité afin d'apprécier s'il est original ou pas. Au contraire, la protection par le droit d'auteur ne s'étend qu'à ce qui est original. L'exercice consistera dès lors plutôt à distinguer, au sein même du programme, ce qui est original de ce qui ne l'est pas.

45. Traditionnellement, à défaut d'une définition dans la loi générale sur le droit d'auteur, l'originalité est décrite comme étant l'empreinte de la personnalité de l'auteur⁵⁷. Selon la jurisprudence de la Cour de cassation, pour être originale, une œuvre doit être « *marquée par la personnalité de son auteur* »⁵⁸. D'autre part, la Cour de cassation présente également l'originalité comme étant « *l'expression de l'effort intellectuel de celui qui l'a réalisée* »⁵⁹. Les deux définitions proposées diffèrent en ce que la première insiste sur l'aspect subjectif de création alors que la seconde met l'accent sur le travail

⁵⁵ Voir article 1^{er}, alinéa 2 de la Directive : « *les idées et principes qui sont à la base de quelque élément que ce soit d'un programme d'ordinateur, y compris ceux qui sont à la base de ses interfaces, ne sont pas protégés par le droit d'auteur en vertu de la présente directive* ». Voir également le considérant 14 de la Directive : « *en accord avec ce principe du droit d'auteur, les idées et principes qui sont à la base de la logique, des algorithmes et des langages de programmation ne sont pas protégés en vertu de la présente directive* ».

⁵⁶ Sur cette notion, voy. A. STROWEL, « L'originalité en droit d'auteur : un critère à géométrie variable », *J.T.*, 1991, p. 513-518 ; F. GOTZEN « Het hof van cassatie en het begrip « oorspronkelijkheid » in het Belgische auteursrecht », *Computerr.*, 1990/4, p. 161.

⁵⁷ A. BERENBOOM, *Le nouveau droit d'auteur et les droits voisins*, Bruxelles, Larcier, 1995, p. 48.

⁵⁸ Voir par exemple : Cass., 25 octobre 1989, *Pas.*, 1990, I, p.239 ; Cass., 27 avril 1989, *Pas.*, 1989, I, p. 908 ; Cass., 2 mars 1993, *Pas.*, 1993, p. 234.

⁵⁹ Cass., 27 avril 1989, *Pas.*, 1989, I, p. 908.

intellectuel, conception se voulant plus objective. Pour une certaine doctrine, ces deux définitions n'expriment en fait qu'une seule exigence, à savoir «*que l'œuvre porte en elle la marque d'une élaboration par une personne*», seul élément objectif perceptible⁶⁰. Pour d'autres auteurs, cette unicité de critère n'est pas si évidente, et la question reste de savoir s'il faut appliquer l'un, l'autre ou les deux, et ce, y compris en tenant compte de la nature de l'œuvre en question⁶¹.

46. En matière de protection des programmes informatiques, le législateur précise la notion d'originalité à l'article 2 de la LPO : «*Un programme d'ordinateur est protégé s'il est original, en ce sens qu'il est une création intellectuelle propre à son auteur. Aucun autre critère ne s'applique pour déterminer s'il peut bénéficier d'une protection par le droit d'auteur.* »⁶².

De cette définition, il nous apparaît qu'en matière de programmes d'ordinateur, la conception subjectiviste tend à s'effacer devant le critère de « création intellectuelle propre à l'auteur ». Cette tendance est confirmée par une jurisprudence récente⁶³.

⁶⁰ F. DE VISSCHER, et B. MICHAUX, *op. cit.*, p. 15.

⁶¹ Voir A. STROWEL, et E. DERCLAYE, *op. cit.*, p. 26.

⁶² Cette définition provient de l'article 1^{er} al. 3 de la Directive, dont le 8^{ème} considérant ajoute encore que «*les critères appliqués pour déterminer si un programme d'ordinateur constitue ou non une oeuvre originale ne devraient comprendre aucune évaluation de la qualité ou de la valeur esthétique du programme* ».

⁶³ «*Eigen intellectuele schepping van de auteur* ». Voir entre autres : Anvers (1^{ère} ch.) 5 juin 2000, A.J.T., 2000-2001, p. 919, note I. VERNIMME ; Civ. Turnhout, 12 septembre 1994, *Computerr.*, 1995/2, p. 65, note M. TAYMANS ; Corr. Hasselt, 16 février 1999, *précité* ; S. DUSOLIER, « Droits d'auteur », in Y. POULLET, et alii, *Droit de l'informatique et des technologies de l'information : Chronique de Jurisprudence 1995 – 2001*, Larcier, p. 51 et s. Dans une décision récente, le président du tribunal de première instance de Bruxelles a donné raison à la partie demanderesse avançant que : «*een computerprogramma, krachtens art. 2 van de Wet van 30 juni 1994 houdende omzetting in Belgisch recht van de Europese Richtlijn van 14 mei 1991 betreffende de rechtsbescherming van computerprogramma's ... bescherming geniet indien het oorspronkelijk is. Dit is het geval van zodra het programma (en voorbereidend materiaal) het resultaat is van de eigen intellectuele inspanning van de auteur, zonder dat wordt vereist dat het werk ook origineel is in de zin van de « algemene » Auteurswet van 30 juni 1994* ». Le juge conclura : «*dat eiseres op overtuigende wijze aantoont dat het concept [...] niet toevallig werd ontwikkeld, maar het resultaat is van langdurige en kostelijke Research & Development, dus van creatief denkproces, een eigen intellectuele schepping van de auteur in de zin van de Softwarewet.* » Prés. Civ. Bruxelles, 30 juin 2003, *précitée*, p. 182. L'on peut donc ainsi pressentir une évolution jurisprudentielle tendant vers la différenciation claire et complète entre le critère d'originalité au regard de la LDA et la « création intellectuelle propre » selon la LPO. Voir dans le même sens Corr. Hasselt, 16 février 1999, *précité*.

Dans leur étude sur la typologie des œuvres en fonction du critère de l'originalité, distinguant les œuvres artistiques, factuelles et fonctionnelles, Alain Strowel et Estelle Derclaye classent les logiciels dans cette troisième catégorie⁶⁴. Les logiciels sont avant tout conçus afin de faire accomplir certaines tâches par une machine et leur valeur s'apprécie eu égard à leur aptitude à remplir cette fonction de façon efficace. C'est donc moins l'« empreinte de la personnalité de l'auteur », notion qui perd une partie de son sens parlant de logiciel⁶⁵, que le travail intellectuel fourni par cet auteur qui sera pris en compte pour apprécier l'originalité d'une œuvre informatique⁶⁶. S'il est vrai qu'on peut prétendre qu'un programme portera la « griffe » de son auteur, dans le sens où un programmeur perspicace parviendra à déterminer avec plus ou moins de succès lequel de ses collègues est à l'origine de tel ou tel code, il est cependant plus difficile d'avancer que les goûts artistiques, l'humeur ou encore l'état d'esprit se refléteront au travers de la programmation. Ce sera plus une question de différence au niveau de la technique ou de la conception logique.

47. Si l'originalité consiste plus en la marque d'un apport intellectuel, qu'entend-t-on dès lors par le fait que l'effort intellectuel doit être « propre » à l'auteur du programme? Cette précision peut être pertinemment illustrée par un passage de l'arrêt de la Cour de cassation française du 7 mars 1986 : « *un effort personnalisé allant au-delà de la simple mise en œuvre d'une logique automatique et contraignante (...) la matérialisation de cet effort résidant dans une structure individualisée* »⁶⁷. Le travail intellectuel de l'auteur doit dès lors aller au-delà des contraintes et des lieux communs de la programmation, du simple « savoir-faire »⁶⁸, et refléter les choix, l'organisation, la structure ou encore la technique propre de l'auteur. En d'autres termes, les lignes de code imposées par la technique, ainsi que par toute autre contrainte externe, ne seront pas considérées comme étant originales, l'auteur n'ayant aucune possibilité de choix.

48. Les logiciels libres n'échappent pas à la règle, et pour qu'ils soient protégés, leurs auteurs devront démontrer qu'ils ont effectué une « création intellectuelle propre ». L'élaboration et l'évolution d'un logiciel libre se distinguent souvent de celles d'un logiciel propriétaire par le nombre de ses

⁶⁴ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 36.

⁶⁵ Voir A. LUCAS, *Droit d'auteur et numérique*, Paris, Litec, 1998, p. 35 : « *La définition subjective traditionnelle cadre mal avec le caractère technique des logiciels, qui sont avant tout des outils s'adressant à la machine et non à l'homme, comme cela apparaît clairement dans le « code-objet », c'est-à-dire en langage machine.* »

⁶⁶ J.-P. TRIAILLE, « La protection des logiciels en Belgique » in *L'informatique et le droit d'auteur*, Cowansville, Yvon Blais, 1990, p. 102.

⁶⁷ Cass. fr. (Ass. Plén.), 7 mars 1986, *JCP* 1986, II, 20631, note J.-M. MOUSSERON, B. TEYSSIE et M. VIVANT; *RIDA* juillet 1986, n°129, p. 136, note A. LUCAS.

⁶⁸ A. LUCAS, *op. cit.*, p. 20 et s.

participants et la diversité de leurs interventions. Afin d’être considéré comme auteur et bénéficiaire de droits sur l’œuvre en cette qualité, chacun de ces participants devra dès lors démontrer que son apport est original. Cette appréciation se fait bien entendu au cas par cas, mais l’on peut déjà imaginer quelques situations typiques qui ne poseront pas trop de problèmes d’appréciation.

Ainsi, le programmeur qui sera créateur du code originaire d’un programme n’aura certainement aucun mal à démontrer le caractère original de son œuvre. Le programmeur qui y aura apporté ses propres lignes de codes afin, par exemple, de pourvoir le logiciel de nouvelles fonctions aura sans doute la même aisance à démontrer que son apport est issu d’un effort intellectuel propre.

A l’inverse, le participant n’ayant que testé le logiciel en y répertoriant les bugs, ne bénéficiera pas de droits d’auteur sur le logiciel, quelle que soit l’assiduité de son travail. De même, l’on peut d’ores et déjà douter que le participant qui n’aurait effectué qu’un travail de « débuggage » voie son apport qualifié d’original⁶⁹. Le doute s’installera cependant si, en plus d’un simple « débuggage », le participant reformule une partie du code.

Section 2 : Titularité

49. Une fois en présence d’une œuvre mise en forme et originale, il faut encore déterminer quel(s) est (sont) le(s) titulaire(s) des droits d’auteur y afférents.

La détermination des titulaires est essentielle dans la mesure où ce sont ces derniers qui bénéficient des droits, prérogatives et protections accordés par le droit d’auteur. En pratique, s’agissant de logiciel libre, ce seront les titulaires des droits qui, entre autres, choisiront la licence sous laquelle sera distribuée le logiciel et auront l’occasion de poursuivre toute personne ne respectant pas les termes de cette dernière.

La règle générale est que la personne physique qui a créé l’œuvre est titulaire des droits sur cette œuvre (§1). Il est cependant possible que plusieurs personnes puissent prétendre détenir des droits d’auteurs sur une œuvre (§2). Des contrats de travail, de commande ou de transfert de droits peuvent intervenir dans le développement des logiciels libres et il s’agira dès lors d’en tenir compte

⁶⁹ Bien qu’il apparaisse, d’après une certaine jurisprudence française, que la qualité de coauteur puisse être reconnue à une personne en charge de la correction : voir A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 163.

dans la détermination des titulaires de droits (§3). Enfin, la loi prévoit également une règle de présomption de titularité (§4).

Sous-section 1 : Personne physique créatrice de l'œuvre

50. La LPO ne prévoyant rien quant à la détermination de l'auteur originaire, il y a lieu de se référer à l'article 6, alinéa 1^{er}, de la loi générale : « *Le titulaire originaire du droit d'auteur est la personne physique qui a créé l'œuvre* ».

C'est donc la personne physique qui a posé l'acte de création intellectuelle propre qui sera investie des droits d'auteurs. Cette règle a pour principale conséquence qu'en droit belge, une personne morale ne pourra jamais être considérée comme étant l'auteur originaire d'une œuvre et ne pourra prétendre à la titularité de droits d'auteur que par l'effet d'un transfert de ceux-ci⁷⁰.

51. Ce principe n'entraîne cependant pas que ce soit systématiquement la personne qui « met la main à la pâte » qui soit forcément titulaire des droits. Il faut en effet toujours tenir compte du critère d'originalité et, dès lors, déterminer qui fait effectivement l'effort intellectuel *créatif* consistant en la détermination de la mise en forme. Un maçon « exécutant » ne deviendra pas titulaire de droits d'auteur sur une œuvre d'architecture, ceux-ci revenant de droit à l'architecte qui aura conçu « intellectuellement » le bâtiment. De même, une personne se faisant dicter les lignes de code d'un programme informatique n'en deviendra pas l'auteur. Cependant, à l'inverse, ne bénéficie également pas de la qualité d'auteur la personne qui se contentera de donner des idées de programmes ou des directives quant à sa réalisation⁷¹, l'idée n'étant jamais en soi protégée par le droit d'auteur. L'on constate donc l'existence de personnes, en amont et en aval de l'auteur, qui ne bénéficient pas de cette qualité⁷². En amont, il va falloir distinguer les fournisseurs d'idées et de directives générales des collaborateurs apportant une participation créative plus importante et participant activement à la « mise en forme » de l'œuvre ; en aval, il faudra distinguer les programmeurs suivant des ordres à la lettre des intervenants ayant un certain degré d'autonomie.

52. Rien, en matière de logiciels libres, ne déroge à ce principe. Ne peut être titulaire d'un logiciel libre que la personne physique qui fait une création

⁷⁰ A noter qu'en droit français, il est possible qu'une personne morale soit titulaire originaire de droits d'auteur en cas d'œuvre collective, notion inexistante en droit belge. Les cas de titularité originaire par une personne morale sont expressément permis par l'article 2, al. 1^{er} de la Directive. Voir par ex. A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 176 et s.

⁷¹ Voir par ex. Anvers (1^{ère} ch.), 5 juin 2000, *précitée*, p. 920.

⁷² A. STROWEL et E. DERCLAYE, *op. cit.*, p. 42.

intellectuelle propre. Ainsi, le participant à un projet libre qui se borne à donner des lignes directrices à un autre programmeur, ou qui ne fait que des remarques ou commentaires sur le logiciel en développement ne sera pas considéré comme auteur. A l'inverse, le participant qui envoie sur une « mailing list » relative à un projet des lignes de codes qu'il aurait personnellement écrites afin d'améliorer ledit projet pourra prétendre bénéficier de droits sur l'œuvre en question, pour autant que ses lignes soient reprises⁷³ dans le code du logiciel.

Sous-section 2 : Multiplicité d'auteurs

53. Si l'article 6 de la LDA prévoit que le titulaire originaire du droit d'auteur est la personne physique qui a créé l'œuvre, la loi n'empêche certainement pas une multiplicité de titulaires de droits.

Par multiplicité d'auteurs, nous entendons le fait que plusieurs auteurs puissent revendiquer le respect de leurs droits sur une œuvre déterminée.

Dans un premier point, nous exposerons la théorie générale en matière de multiplicité d'auteurs. Nous appliquerons ensuite cette théorie aux logiciels libres dans un second point.

§ 1. Théorie générale

54. En droit belge, deux cas de figure peuvent se présenter : l'œuvre de collaboration et l'œuvre dérivée. Afin d'être exhaustif et en vue d'élargir notre perspective, nous tiendrons également compte d'un troisième cas prévu exclusivement en droit français, à savoir l'œuvre collective. En outre, le droit français nous sera utile à plus d'un titre, car le code français de la propriété intellectuelle (CPI) fournit, à l'inverse de la loi belge, une définition précise de chacune de ces notions.

A. Œuvre de collaboration

55. En son article 2, alinéa 1^{er}, la Directive envisage explicitement les situations où un groupe de personnes physiques crée le programme et où ce groupe doit dès lors être considéré comme étant son auteur. Ce genre de création de groupe est appelé, en droit belge et français, œuvre de collaboration.

⁷³ Cette reprise peut être « telle quelle » ou comporter des modifications, du moment que l'apport original de l'auteur à la mise en forme d'expression de l'œuvre soit identifiable.

Vu l'inexistence de définition légale de la collaboration en droit belge, on se référera à la définition française⁷⁴. Le CPI, en son article L. 113-2, la définit comme étant : « *l'œuvre à la création de laquelle ont concouru plusieurs personnes physiques* ».

1) Eléments constitutifs

56. Nous retrouvons dans cette définition trois critères à remplir afin qu'une œuvre soit considérée comme étant une œuvre de collaboration :

a) « Plusieurs personnes physiques »

57. L'œuvre doit être réalisée par deux personnes au moins. En concordance avec l'article 6 de la LDA et la conception personnaliste du droit d'auteur, ces personnes doivent obligatoirement être des personnes physiques.

b) « Création »

58. Chacune de ces personnes physiques doit faire un apport créatif original et participer à la mise en forme de l'œuvre par sa contribution personnelle. Autrement dit, la part de travail des coauteurs doit répondre à la condition d'originalité telle qu'exposée ci-dessus. Le simple apport d'idées et/ou la simple mise en forme suivant les ordres précis de l'auteur ne seront pas considérés comme étant des collaborations originales. Le simple apport d'information, dans le cadre du métier de documentaliste, ne constitue pas un apport original⁷⁵. Ces critères feront dès lors l'objet d'une appréciation au cas par cas⁷⁶. S'agissant de programmation, la règle des œuvres de collaboration s'applique lorsque les collaborateurs sont intervenus au « stade de la formalisation du logiciel »⁷⁷. Dans ce sens, « *l'obligation de collaborer avec le fournisseur qui pèse sur le client ne suffit pas à faire de celui-ci un coauteur, dès lors qu'il ne participe pas directement à la mise en forme du logiciel* »⁷⁸.

⁷⁴ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 42, n°35.

⁷⁵ M. BUYDENS, « La bande dessinée comme œuvre de collaboration », in *Droit d'auteur et bande dessinée*, Bruxelles, Bruylant, L.G.D.J., 1997, p. 13.

⁷⁶ Pour des exemples d'appréciation au cas par cas en droit d'auteur général, voir par exemple, A. STROWEL et E. DERCLAYE, *op. cit.*, p. 42 et s. ; A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 161 et s.

⁷⁷ A. BENSOUSSAN, *op. cit.*, p. 71.

⁷⁸ A. LUCAS, et H.-J. LUCAS, *op. cit.*, p. 161.

L'existence de cet apport créatif à la mise en forme originale est le seul critère pertinent afin d'apprécier si le collaborateur doit ou non être considéré comme coauteur. La qualité ou la quantité du code informatique apporté par le collaborateur, la fonction de ce dernier, la nature ou le genre de cet apport n'influenceront généralement pas cette appréciation. Les coauteurs ne doivent pas être sur un pied d'égalité⁷⁹ au regard de la quantité ou la qualité de leurs apports.

59. En ce qui concerne la répartition des droits, jurisprudence française à l'appui⁸⁰, Alain Berenboom fait une distinction parmi les participants à la création d'une œuvre indivise⁸¹, entre les coauteurs titulaires de droits d'auteur sur l'ensemble de l'œuvre elle-même et les créateurs accessoires qui n'ont de droits que sur leur seule création personnelle⁸². Cette position est cependant loin de faire l'unanimité, une grande partie de la doctrine récusant expressément cette notion⁸³.

S'il est vrai qu'en soi cette distinction n'est issue que d'une jurisprudence accidentelle et d'une doctrine minoritaire, elle se comprend dans la mesure où les droits acquis sur l'œuvre ne sont pas proportionnels à la contribution apportée, ce qui, dans des situations extrêmes, peut choquer. Il nous semble cependant qu'une juste interprétation du concept de collaboration, et particulièrement de la notion de concours qu'il implique, permet d'arriver au même résultat, à savoir la possibilité de distinguer les collaborateurs véritables et les auteurs d'œuvres *intégrées* à l'œuvre de collaboration.

c) « Concours »

60. Enfin, ces personnes physiques doivent avoir concouru à la création de l'œuvre. Cette situation de concours fut également qualifiée d'intimité spirituelle⁸⁴, d'inspiration commune, de travail de concert⁸⁵, de mutuel

⁷⁹ A. LUCAS, *op. cit.*, p. 93.

⁸⁰ Paris, 11 mai 1965, *J.T.*, 1965, p. 465.

⁸¹ A savoir, une œuvre unique dont les éléments originaux ont été apportés par plusieurs auteurs.

⁸² A. BERENBOOM, *op. cit.*, art. 144.

⁸³ L. VAN BUNNEN, « Examen de Jurisprudence (1978 à 1981) – Droit d'auteur – Dessins et modèles », *R.C.J.B.*, 1984, p.332 ; J. CORBET, « Aperçu de la jurisprudence belge récente », *R.I.D.A.*, Octobre 1982, p. 3 et s. ; F. BRISON, « Le titulaire du droit d'auteur », *D.A.O.R.*, 1992, p. 105 ; M. DEMEUR, « Le Dante aux petites cuillères : œuvre des arts plastiques et collaboration », note sous Cass., 22 mai 1980, *Ing. Cons.*, 1981, p. 363 ; A. PUTTEMANS, « Droit d'auteur : indivision et œuvres de collaboration » in *Les copropriétés*, Bruxelles, Bruylant, 1999, p. 356 et s.

⁸⁴ Paris, 11 mai 1965, *J.T.*, 1965, p. 465.

contrôle⁸⁶,... Autant de termes pour décrire une situation où les auteurs ont la volonté de créer une œuvre commune, se concertent et unissent leurs efforts à cette fin. Il n'est pas nécessaire qu'une vraie coopération soit mise sur pied, mais qu'il existe un projet commun auquel tous les coauteurs soient associés⁸⁷. Selon Jan Corbet, les coauteurs doivent également se connaître, avoir été en contact et s'être l'un l'autre influencés⁸⁸.

D'après la Cour d'appel de Paris, pour qu'un participant soit vraiment considéré comme collaborateur coauteur de l'œuvre, il faut qu'il ait participé à la création du « *tout indivisible qui se suffit à lui même et a une valeur propre* »⁸⁹. Il nous semble que pour bénéficier de la qualité de coauteur, il faut donc que le collaborateur ait une influence et une emprise sur la globalité de l'œuvre, autrement dit, sur ce tout « indivisible ». En d'autres termes, son emprise décisionnelle sur l'œuvre globale doit dépasser celle qu'il a sur son seul apport⁹⁰. Si tel n'est pas le cas, cet apport est, à notre sens, à analyser comme étant une œuvre dont est dérivée l'œuvre de collaboration qui l'intègre⁹¹.

61. Selon une partie de la doctrine, l'apport créatif des différents auteurs doit être simultanément dans l'élaboration d'une même œuvre⁹². Pour d'autres, il faut que

⁸⁵ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 42 ; TGI Paris, 3^{ème} ch., 16 juin 1995 : *Gaz. Pal.* 1996, 1, somm. p. 50.

⁸⁶ H. DESBOIS, *Le droit d'auteur en France*, Dalloz, 3^e éd., 1978, n°133.

⁸⁷ A. LUCAS, *op. cit.*, p. 93.

⁸⁸ J. CORBET, *Auteursrecht*, Bruxelles, Story-Scientia, 1991, p. 19.

⁸⁹ Paris, 11 mai 1965, *précité*. Voir également Paris, 16 janvier 1992, *R.I.D.A.*, avril 1992, p. 204.

⁹⁰ Dans ce sens, nous rejoindrions le point de vue d'Alain Berenboom, non pas sur le fait qu'il faille distinguer parmi les collaborateurs, les coauteurs des créateurs accessoires, mais dans la mesure où, pour bénéficier de la qualité de collaborateur, il s'agirait en fait de vérifier si, sans l'apport de l'auteur considéré, l'œuvre commune aurait été différente (A. BERENBOOM, *op. cit.*, art. 144).

⁹¹ Cette interprétation permet d'appréhender certaines situations ambiguës. Prenons par exemple la conception d'un décor représentant un château pour une représentation théâtrale. Ce décor peut très bien avoir été conçu sur commande par un artiste, sans que celui-ci ait même la moindre idée du scénario ou du style de la pièce présentée. Dans ce cas, cet artiste ne peut être qualifié de collaborateur et dès lors se voir attribuer un droit sur l'ensemble de la pièce : il sera auteur de son œuvre (le décor) qui sera intégrée, avec son autorisation, à la pièce. Ce seront dès lors bien les règles en matière d'œuvre dérivée qui s'appliqueront. A l'inverse, l'on peut imaginer que l'artiste en question se voie remettre le script de la pièce, en ayant pour mission de créer un château qui corresponde à son ambiance et son style. Dans ce cas d'espèce, l'on retrouve bien les éléments d'inspiration commune ou de travail de concert.

⁹² « *Samenwerking veronderstelt tevens een gelijktijdige creatieve inbreng bij de opbouw van een zelfde werk. Wie op een voorafbestaand werk eerst later een creatieve inbreng uitoefent verwerft eveneens een auteursrecht, maar wel met een verschillend statuut, nl. dat van de adaptatie* ». F. GOTZEN, *Auteursrecht*, Syllabus voortgezette academische opleiding intellectuele rechten, année 2002-2003, p. 29.

« *les contributions rassemblées obéissent à une inspiration commune, qu'il y ait concertation entre les auteurs et non simple juxtaposition des contributions* »⁹³. D'après nous, ces deux énoncés, à première vue différents, expriment en fait le même critère que l'on pourrait décrire comme « un critère de l'unicité de l'œuvre ». Alors que les premiers énoncent la simultanéité dans l'apport créatif, il ne faut pourtant pas que les contributions soient effectuées en même temps⁹⁴, ni au même endroit⁹⁵. Il s'agit plutôt d'établir la différence entre, d'une part, une collaboration, c'est-à-dire le rassemblement de contributions originales dans une seule œuvre, et d'autre part, une adaptation, c'est-à-dire la reprise par un auteur d'une œuvre achevée par un autre auteur, afin de créer une deuxième œuvre, sur laquelle le premier auteur bénéficiera également de droits d'auteur. Lorsque l'on insiste sur l'inspiration commune, c'est pour distinguer l'œuvre issue d'un projet commun d'une juxtaposition d'œuvres d'auteurs uniques.

2) Collaborations divisibles et indivisibles

62. La Directive prévoit en son article 2, alinéa 2, que : « *Lorsqu'un programme d'ordinateur est créé en commun par plusieurs personnes physiques, les droits exclusifs sont détenus en commun par ces personnes* ». Une des conséquences de ce principe est que la période de 70 ans de protection démarre à partir du 1^{er} janvier qui suit le décès du dernier coauteur survivant (art. 2, §2, alinéa 1^{er} LDA).

A la lecture des articles 4 et 5 de la LDA, il y a cependant lieu de faire la distinction entre deux types de collaborations : les collaborations indivisibles et les collaborations divisibles.

Notons d'emblée que cette distinction n'affecte pas le caractère unique du « tout indivisible » de l'œuvre de collaboration. Il s'agit plus ici de déterminer dans quelle mesure les droits d'auteur relatifs à l'œuvre sont indivis ou non⁹⁶.

a) Collaboration indivisible

63. Une collaboration indivisible est une œuvre de collaboration où les contributions des auteurs ne peuvent être individualisées ou distinguées. L'exemple typique de l'œuvre de collaboration indivisible est l'article rédigé par

⁹³ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 44, n°36.

⁹⁴ Les contributions peuvent en effet être successives. Voy. A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 164.

⁹⁵ A. BERENBOOM, *op. cit.* p. 145.

⁹⁶ En ce sens, voir A. PUTTEMANS, 1999, *op. cit.* p. 358.

plusieurs auteurs. En matière d'informatique, la situation classique sera la participation de plusieurs programmeurs à l'écriture du code d'un même fichier.

En cas d'œuvre de collaboration indivisible, la loi organise un système d'indivision rudimentaire.

64. L'article 4, alinéa 1^{er}, de la LDA prévoit que « *lorsque le droit d'auteur est indivis, l'exercice de ce droit est réglé par les conventions. A défaut de conventions, aucun des auteurs ne peut l'exercer isolément, sauf aux tribunaux à se prononcer en cas de désaccord* ». Les droits d'auteur ne peuvent donc s'exercer sans l'accord de tous les coauteurs, chacun d'eux jouissant de droits égaux sur l'œuvre, sauf si ces coauteurs ont prévu le contraire dans une convention. Si aucune convention n'est signée et si les coauteurs ne s'accordent pas, le législateur s'en remet à la sagesse du juge. Dans son article 4, alinéa 3, la LDA ajoute que « *les tribunaux pourront toujours subordonner l'autorisation de publier l'œuvre aux mesures qu'ils jugeront utiles de prescrire; ils pourront décider à la demande de l'auteur opposant, que celui-ci ne participera ni aux frais, ni aux bénéfices de l'exploitation ou que son nom ne figurera pas sur l'œuvre* ».

65. En ce qui concerne la poursuite des contrefaçons et la violation des droits d'auteur, l'article 4, alinéa 2 précise toutefois que « *chacun des auteurs reste libre de poursuivre, en son nom et sans l'intervention des autres, l'atteinte qui serait portée au droit d'auteur et de réclamer des dommages et intérêts pour sa part* ». Comme le soulignent Fernand De Visscher et Benoît Michaux, cette disposition permet avant tout aux coauteurs de contrecarrer l'autorisation qui aurait été donnée à leur insu par les autres coauteurs⁹⁷.

b) Collaboration divisible

66. Une collaboration divisible est une œuvre de collaboration dans laquelle la contribution des auteurs peut être individualisée. Un exemple classique de l'œuvre de collaboration divisible est la bande dessinée, dont l'un des coauteurs a créé les planches et l'autre le scénario et les textes. En matière informatique, l'on pourrait imaginer par exemple qu'un programmeur crée une structure de programme et qu'un autre crée un fichier contenant certaines fonctionnalités qui seront appelées par le programme (fichier principal).

⁹⁷ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 40.

67. Le régime de la collaboration divisible est identique à celui de la collaboration indivisible⁹⁸, si ce n'est que, sous certaines conditions, le législateur admet que les coauteurs puissent exploiter isolément leur contribution. Par exemple, l'auteur des textes d'une chanson peut éditer ceux-ci sous forme de poèmes.

L'article 5 de la LDA prévoit cependant certaines restrictions à ce droit : *« lorsqu'il s'agit d'une oeuvre de collaboration où la contribution des auteurs peut être individualisée, ces auteurs ne peuvent, sauf convention contraire, traiter de leurs oeuvres avec des collaborateurs nouveaux. Néanmoins, ils auront le droit d'exploiter isolément leur contribution, pour autant que cette exploitation ne porte pas préjudice à l'oeuvre commune »*.

Cette exploitation isolée peut donc être effectuée à deux conditions :

- elle ne porte pas atteinte à l'exploitation de l'oeuvre commune ; et
- elle ne consiste pas en la « réutilisation » de la contribution dans une autre oeuvre de collaboration.

Ces deux restrictions sont analysées comme étant des obligations de non-concurrence : l'exploitation isolée d'une contribution ne peut pas porter atteinte à l'exploitation de l'oeuvre pour laquelle elle a été créée⁹⁹.

3) Œuvres audiovisuelles

68. Les articles 14 à 20 de la LDA prévoient un régime spécifique pour les oeuvres de collaboration particulières que sont les oeuvres audiovisuelles. Les oeuvres audiovisuelles ne sont pas définies dans la LDA. Par contre, l'article L 112 – 2 du CPI français les définit comme étant des séquences animées d'images, sonorisées ou non. De façon générale, la doctrine belge retient cette définition et considère que sont classés dans cette catégorie non seulement les films et dessins animés, mais encore les jeux vidéo, les clips musicaux et les montages de diapositives¹⁰⁰.

Le régime spécifique des oeuvres audiovisuelles concerne les logiciels dans la mesure où, à l'ère du multimédia et du numérique, de plus en plus de ces oeuvres reposent sur des bases informatiques. La question de la qualification du

⁹⁸ Nous renvoyons dès lors à ce qui a été dit au point précédent.

⁹⁹ A. STROWEL, et E. DERCLAYE, *op. cit.*, p. 45 et s.; F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 40.

¹⁰⁰ A. BERENBOOM, *op. cit.*, p. 192 et s.; F. DE VISSCHER, et B. MICHAUX, *op. cit.*, p. 210 et s. ; A. STROWEL et E. DERCLAYE, *op. cit.*, p. 90 et s.

multimédia fait l'objet d'importants développements doctrinaux. Nous ne nous étendrons pas sur cette question¹⁰¹, mais nous nous contenterons de constater qu'en Belgique, un jeu vidéo s'est vu gratifié par la jurisprudence d'une double qualification¹⁰² : alors que le code du logiciel est protégé en tant que programme d'ordinateur, l'interface visuelle qui fait apparaître le déroulement du jeu vidéo à l'écran est protégé en tant qu'œuvre audiovisuelle¹⁰³.

69. Le régime organisé repose sur plusieurs principes :

- certaines présomptions de titularité ont été instaurées (art. 14 LDA). Outre le réalisateur principal, qui est d'emblée – et de façon irréfragable – coauteur, certains collaborateurs clés bénéficient d'une présomption réfragable de coauteurs : l'auteur du scénario, l'auteur de l'adaptation, l'auteur des textes, l'auteur graphique de l'animation, l'auteur des compositions musicales spécialement réalisées pour l'œuvre et en cas de « remake » les auteurs de l'œuvre originale si leur contribution est reprise. Ces présomptions ont également pour effet d'éviter les problèmes de preuve en matière d'appréciation de l'originalité des apports créatifs ;
- le droit moral de divulgation a été adapté. Ainsi, sous certaines conditions, le réalisateur pourra utiliser la contribution inachevée d'un des coauteurs ou créateurs non-titulaires (art. 15 LDA). De même, seuls le réalisateur et le producteur décident si l'œuvre est achevée ou non (art. 16 LDA) (*infra*, n° 161 et s.) ;
- les œuvres audiovisuelles bénéficient également de règles contractuelles spéciales (art. 17 à 20 LDA). En particulier, l'article 17 de la LDA prévoit, sauf convention contraire et à l'exception des droits des auteurs de compositions musicales, une présomption de cession des droits d'exploitation au producteur.

Dans le cadre de la présente analyse sur la titularité des logiciels, c'est le premier principe qui retiendra notre attention.

¹⁰¹ Voir entre autres la quatrième partie de l'ouvrage précité d'Alain STROWEL et Estelle DERCLAYE consacrée à cette question, *op. cit.*, p. 357 et s.

¹⁰² Décision *Nintendo c. Horelec*, Prés. Civ. Bruxelles, 12 décembre 1995, *I.R.D.I.*, 1996, p. 89, confirmé par Bruxelles (9^{ème} Ch.), 11 avril 1997, *A & M*, 1997, p. 265.

¹⁰³ A. STROWEL, et E. DERCLAYE, *op. cit.*, p. 366 et s.

B. Œuvre dérivée ou composite

70. La LDA ne donnant pas de définition de l'œuvre dérivée (ou composite), c'est à nouveau dans le Code de Propriété Intellectuelle français qu'on la trouvera. Son article L113-2, définit l'œuvre dérivée comme étant « *l'œuvre nouvelle à laquelle est incorporée une œuvre préexistante sans la collaboration de l'auteur de cette dernière* ». Le CPI précise dans son article 113-2 que « *l'œuvre composite est la propriété de l'auteur qui l'a réalisée, sous réserve des droits de l'auteur de l'œuvre préexistante* ».

71. L'œuvre dérivée ou composite est l'œuvre dont l'originalité est relative, dans la mesure où elle dépend d'une œuvre préexistante dont elle reprend des éléments originaux¹⁰⁴. En d'autres termes, l'œuvre dérivée est une œuvre distincte de l'œuvre originale qui lui est incorporée en tout ou en partie. Cette incorporation n'est légale que si l'auteur de l'œuvre dérivée demande l'autorisation à l'auteur de l'œuvre originale d'en reprendre des éléments, et d'exploiter l'œuvre résultante¹⁰⁵. En effet, sans cette autorisation préalable, la réalisation et l'exploitation de l'œuvre dérivée porteraient atteinte aux droits patrimoniaux de l'auteur de l'œuvre initiale¹⁰⁶. L'auteur de l'œuvre initiale conserve donc ses droits d'auteur sur celle-ci ainsi que sur les éléments originaux qui ont été empruntés dans l'œuvre dérivée.

¹⁰⁴ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 41.

¹⁰⁵ Ainsi par exemple, si nous voulons incorporer une mélodie originale issue d'un morceau musical dont nous ne sommes pas l'auteur dans notre propre composition, que nous comptons par ailleurs enregistrer, distribuer et jouer en public, nous devons en demander l'autorisation au titulaire des droits sur cette mélodie.

¹⁰⁶ L'on peut se poser la question de savoir si le simple fait de créer l'œuvre dérivée requiert l'autorisation préalable de l'auteur de l'œuvre originale. Voir H. DESBOIS, *op. cit.*, n°613 ; A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 189 et s. Il nous semble que la réponse à cette question se trouve dans l'articulation entre les droits exclusifs de l'auteur et les exceptions à ces droits. L'adaptation étant une forme de reproduction, l'autorisation préalable ne sera pas requise si l'acte de reproduction est couvert par une exception. Ainsi André et Henri-Jacques Lucas estiment que la plupart des adaptations sont, au stade de projet non encore publié, couvertes par le droit de copie privée. Cette solution est certainement applicable en droit belge, mais il nous faut d'emblée soulever deux restrictions importantes. Tout d'abord, sur le plan du droit d'auteur général, l'exception de copie privée ne couvre que les actes posés dans le cercle de la famille et pour le cercle de la famille. Par exemple, un acte de reproduction fait sur le lieu de travail avec des instruments de travail ne seront déjà plus couverts par l'exception. Deuxièmement, en matière informatique, la copie privée ne fait pas partie des exceptions aux droits exclusifs des programmeurs. Dès lors, il nous semble qu'il faudra toujours obtenir l'autorisation préalable de l'auteur du programme original avant de créer une œuvre informatique dérivée. Bien entendu, il est plus sage d'éviter la question par la pratique systématique des contrats d'adaptation préalables à la création. Voir F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 43.

Cependant, à l'inverse de l'œuvre de collaboration, l'œuvre dérivée n'est pas le fruit d'un travail commun, d'une intimité spirituelle entre son auteur et l'auteur de l'œuvre première. Ce dernier n'a pas participé à la création de l'œuvre dérivée, et n'a aucun droit sur cette œuvre en tant que telle. Les droits d'auteur sur l'œuvre dérivée appartiennent seulement à son auteur¹⁰⁷. Dès lors, lorsque l'on parle de multiplicité d'auteurs en matière d'œuvre dérivée, c'est uniquement pour mettre en exergue l'autorisation préalable de l'auteur de l'œuvre originale nécessaire à la création et à l'exploitation de l'œuvre dérivée.

72. Au cas où l'œuvre dérivée serait constituée essentiellement d'éléments issus d'œuvres originales existantes, il ne faut pas perdre de vue que l'auteur de l'œuvre dérivée ne bénéficiera de la qualité d'auteur que s'il a réalisé lui-même un apport original.

73. Il faut noter que les règles énoncées ci-dessus sont applicables aux chaînes d'œuvres dérivées. Ainsi, l'auteur d'une œuvre dérivée C, basée sur une œuvre B dérivée de l'œuvre originale A, devra obtenir l'autorisation des auteurs des œuvres A et B dans la mesure où des éléments originaux sont empruntés à ces deux œuvres¹⁰⁸.

C. Œuvre collective

74. L'œuvre collective n'est pas reconnue en Belgique mais bien en France. Selon l'article L113-2 du CPI, « *est dite collective l'œuvre créée sur l'initiative d'une personne physique ou morale qui l'édite, la publie et la divulgue sous sa direction et son nom et dans laquelle la contribution personnelle des divers auteurs participant à son élaboration se fond dans l'ensemble en vue duquel elle est conçue, sans qu'il soit possible d'attribuer à chacun d'eux un droit distinct sur l'ensemble réalisé* ».

L'article L 113-5 du CPI, ajoute : « *l'oeuvre collective est, sauf preuve contraire, la propriété de la personne physique ou morale sous le nom de laquelle elle est divulguée. Cette personne est investie des droits de l'auteur* ».

Par ailleurs, l'article 2, alinéa 1^{er}, de la Directive prévoit expressément la possibilité pour les Etats membres de reconnaître la catégorie des œuvres

¹⁰⁷ Comme nous l'avons vu plus haut, constitue une exception à ce principe la présomption par laquelle, en matière d'œuvre audiovisuelle, les auteurs de l'œuvre originale sont considérés comme coauteurs si leur collaboration est reprise.

¹⁰⁸ A. BERENBOOM, *op. cit.*, p. 104.

collectives¹⁰⁹. Le législateur européen a dès lors accepté l'idée qu'un programme d'ordinateur puisse être qualifié d'œuvre collective.

1) Éléments constitutifs

75. Deux critères doivent être remplis afin qu'une œuvre puisse être qualifiée d'œuvre collective :

a) L'œuvre est créée à l'initiative et sous la direction d'un entrepreneur

76. Une personne, physique ou morale, doit avoir eu l'initiative de l'œuvre, en avoir été le « moteur pendant la phase d'élaboration »¹¹⁰. C'est cette personne qui contrôle le processus de création et qui, au final, bénéficiera seule des droits d'auteur sur l'ensemble de l'œuvre. La tâche de l'entrepreneur est dès lors active : il donne des directives et harmonise les créations des auteurs participants. L'œuvre collective est le seul cas où, en France, une personne morale peut devenir titulaire originaire de droits d'auteur sur une œuvre.

Cette personne doit également éditer, publier et divulguer l'œuvre à son nom¹¹¹. C'est donc seulement le nom de cette personne qui pourra figurer sur l'œuvre. Si le nom des différents auteurs apparaît, il ne pourrait sans doute plus s'agir d'œuvre collective¹¹².

b) Il y a fusion des contributions entraînant l'impossibilité d'attribuer à chaque auteur des droits distincts sur l'ensemble

77. Ce deuxième critère est connu pour être l'un des points les plus obscurs du droit d'auteur français. Ce critère est parfois interprété comme étant l'absence de concertation entre les auteurs participants ou l'impossibilité d'isoler les contributions. On pourrait également y voir un argument prônant la protection de l'investissement fait par le producteur¹¹³.

¹⁰⁹ Il prévoit en effet que : « lorsque les oeuvres collectives sont reconnues par la législation d'un État membre, la personne considérée par la législation de l'État membre concerné comme ayant créé l'oeuvre est réputée en être l'auteur ».

¹¹⁰ A. LUCAS et H.-J. LUCAS, *op cit.*, p. 177 et s.

¹¹¹ Cette énumération semble être cumulative, ce qui posera donc problème pour les types d'œuvres qui n'auront pas vocation à être éditées ou communiquées au public, tels les logiciels. Pour ces raisons, la jurisprudence française récente n'applique plus ce texte à la lettre. A. LUCAS et H.-J. LUCAS, *ibidem*.

¹¹² A noter qu'à notre sens, rien n'empêche que plusieurs personnes soient titulaires de droits indivis sur une œuvre collective.

¹¹³ A. LUCAS et H.-J. LUCAS, *op cit.*, p. 177 et s.

Cette catégorie n'était à l'origine destinée qu'aux encyclopédies, aux dictionnaires et aux journaux. Dans ce sens, on remarque en effet qu'il eut été impossible et ingérable de reconnaître à tout auteur participant à ce genre d'œuvres des droits sur l'ensemble de l'œuvre. Il s'agit donc, à notre sens, d'interpréter ce critère en gardant cette considération d'ordre pratique à l'esprit. La doctrine française admet elle-même que la distinction entre les œuvres de collaboration et les œuvres collectives est parfois très difficile à établir¹¹⁴.

2) Droits sur l'œuvre

78. Seul l'entrepreneur est titulaire des droits moraux et patrimoniaux sur l'œuvre collective. En ce qui concerne l'œuvre collective dans son ensemble, les auteurs participants n'ont aucun droit, si ce n'est de réclamer leur redevance forfaitaire en échange de leur contribution.

Par contre, on peut se demander quels sont les droits dont bénéficient les auteurs participants sur leur contribution.

En matière de journaux et périodiques, le CPI prévoit en son article L. 121-8 que l'auteur a le droit exclusif de réunir ses articles et ses discours en recueil et de les publier ou d'en autoriser la publication sous cette forme. Cet article prévoit également que les auteurs de tels articles peuvent exploiter isolément leur contribution à condition que cette exploitation ne fasse pas concurrence au journal ou au périodique. Il semble que cette règle puisse être, par analogie, appliquée à toute autre contribution à une œuvre collective. De même, les auteurs de ces contributions bénéficient également de droits moraux, qu'ils ne peuvent cependant faire valoir que d'une façon qui ne soit pas contraire à la nature des œuvres collectives¹¹⁵.

§ 2. Application aux logiciels libres

79. Avant d'appliquer la théorie générale exposée ci-dessus aux logiciels libres, il est essentiel de décrire les différentes évolutions que peuvent subir les logiciels libres. Leur développement est, en effet, complexe et les cas de figure sont pratiquement illimités.

¹¹⁴ F. DEGOY-VACHER, « De la subtile distinction œuvre collective/œuvre de collaboration », <http://www.lettresdudroit.com/article.list/de-la-subtile-distinction-oeuvre-collective/-oeuvre-de-coll.htm>. Voy. TGI Paris, 3e ch., 2e sect., 15 mars 2002, Madame Bachetta et al. c/ SA Ellipses Editions Marketing et al., *Juris-Data* n° 2002-170857, *Communication, Commerce Electronique* mai 2002, p. 24.

¹¹⁵ A. LUCAS et H.-J. LUCAS, *op cit.*, p. 177 et s.

Nous appliquerons ensuite les règles de titularité aux logiciels libres en tenant compte des particularités de leur développement. Nous nous poserons la question de savoir, à chaque stade de l'évolution d'un logiciel, si l'on est en présence d'une œuvre dérivée, de collaboration, ou des deux simultanément.

A. Création et développement des logiciels libres

80. Lorsque la création d'un logiciel implique la participation de plusieurs personnes, pour le développement de grands programmes tels que des systèmes d'exploitation par exemple, il existe, d'après Raymond, deux méthodes de développement : l'une est comparable à la construction d'une cathédrale et l'autre à l'activité d'un bazar¹¹⁶. La construction d'une cathédrale est le fruit du génie de quelques architectes qui conçoivent, isolés, la globalité de l'œuvre. L'entièreté du développement est, du début à la fin, sous leur entier contrôle et l'œuvre n'est divulguée que lorsqu'elle est achevée et prête à l'emploi. A l'inverse, le bazar est constitué d'une multitude d'individus qui, seuls ou par petits groupes, développent spontanément des parties de projets selon leurs propres approches et spécialités et s'adressent à tout un chacun afin de résoudre les problèmes rencontrés. Les logiciels traditionnels « propriétaires » sont créés selon le schéma de la cathédrale. Les logiciels libres, dans la plupart des cas, voient le jour au sein d'un gigantesque bazar, grâce au principe de disponibilité et de liberté des codes sources¹¹⁷.

1) Quelques cas de figure

81. Le caractère libre ou non d'un logiciel dépend de la façon dont son ou ses auteur(s) exerce(nt) leurs droits sur cette œuvre. En d'autres termes, la licence définira si le logiciel est libre ou non. Ni la qualité du logiciel, ni sa taille, ni le nombre de participants au projet n'ont d'influence sur cette qualification.

82. Les cas de figure possibles sont infinis. Chaque logiciel a sa vie propre et il est dès lors inutile de vouloir créer un schéma de création et de développement standard.

¹¹⁶ E. S. RAYMOND, *The Cathedral and the Bazaar*, <http://www.catb.org/~esr/writings/cathedral-bazaar/>.

¹¹⁷ M.M. GROENENBOOM, « Softwares licenties : van closed source tot open source », *Computerrecht* 2002/01, p. 26. : « Omdat de broncode beschikbaar is, is iedereen met interesse voor het ontwikkelen van software potentieel programmeur ».

A des fins purement illustratives, nous décrivons ci-dessous quelques situations courantes :

- Un logiciel libre peut être le programme développé par un individu, et sera par exemple « uploadé » sur une page personnelle et offert en téléchargement, inclus dans un « package » ou distribué de la main à la main (ou plutôt de « poste à poste »). Distribué sous licence libre, ce programme vivra sa vie selon un destin imprévisible : il sera laissé pour compte et personne ne s'y intéressera, ou à l'inverse, il sera développé par d'autres programmeurs, locaux ou situés à l'autre bout de la planète, il sera intégré dans d'autres programmes, refondu dans des codes plus importants...
- Un logiciel libre peut également être issu d'un projet de quelques-uns, qui vont s'organiser pour développer un programme commun. Selon leur organisation, ils pourront faire ce développement de façon anarchique, chacun apportant ses idées et ses bouts de code, ou de façon coordonnée. Ils peuvent se répartir les tâches, désigner des responsables pour harmoniser le travail du groupe. Ce projet peut donner naissance à plusieurs versions du programme, dont les améliorations subséquentes auront été apportées par les participants au projet, des amis ou même parfois des tiers au projet. Le sort de ce genre de projet est tout aussi imprévisible, mais il est certain que plus l'effort et l'ingéniosité des participants sont importants, plus le programme est susceptible d'être de qualité et donc d'intéresser les utilisateurs ou d'autres développeurs.
- On peut également citer les gros projets de renommée plus importante, développés par des « professionnels » du libre. A ce niveau, le développement est organisé et contrôlé. L'apport de tout un chacun est souvent apprécié, mais au sein du projet même, les participations extérieures sont soumises à l'aval d'un responsable, avant de pouvoir être intégrées. Le programme développé sera destiné à être distribué au plus grand nombre d'utilisateurs possibles soit seul, soit dans des packages ou des distributions. Le programme fera l'objet de plusieurs versions, *bêtas*¹¹⁸ ou stables. Même si, le programme étant sous licence libre, tout le monde peut le modifier et redistribuer, les responsables du projet auront le contrôle de la sortie de « versions officielles ».

¹¹⁸ Les versions bêtas sont des versions incomplètes ou de « test ».

2) Le développement par « Version Management Systems »

83. Lorsqu'un programme est développé simultanément par plusieurs personnes, l'on peut vite arriver à diverses situations conflictuelles. Par exemple, deux participants modifieront la même partie de programme ou l'un d'entre eux modifiera ou remplacera une fonction, alors qu'un autre rédigera du code faisant appel à celle-ci ou à son résultat... Plus les participants sont nombreux, plus ces situations conflictuelles risquent de voir le jour.

84. Les « Version Management Systems », des programmes de gestion de développement basés sur des sortes de grands registres de modifications, permettent d'éviter ce genre de désagréments. Le logiciel CVS (Concurrent Versions System) est un des standards en la matière¹¹⁹. Outre le fait qu'il règle les conflits entre les modifications contradictoires simultanées, ce logiciel, distribué sous licence G.P.L., permet de gérer l'évolution d'un programme et de toutes ses versions. Le principe est qu'avec ce système, il est possible de suivre toute l'évolution du code, et de retrouver le programme sous toutes ses versions. Lorsque nous parlons de versions, nous ne faisons pas référence aux versions « officielles » du programme, c'est-à-dire aux versions stables et complètes du programme qui sont publiées à chaque fois que les responsables du projet l'estiment pertinent, mais bien à chaque instant où le programme subit une modification causée par quiconque participe au projet et a accès au serveur CVS. Le principe est très simple : chaque participant travaille isolément de son côté (dans son propre « directory » (registre) enregistré à son nom). Le système CVS ne sauvegarde pas l'entièreté du programme à chaque modification, mais seulement les modifications faites au programme, en conservant la date à laquelle ces modifications ont été effectuées¹²⁰. A tout instant, il est possible de fusionner le programme et ses modifications afin d'obtenir aussi bien la version actuelle qu'une version antérieure à chaque moment déterminé du passé.

Pour pouvoir faire partie du projet et avoir la possibilité de modifier le code, il faut donc avoir un compte (ou « directory »). Des participants « étrangers » voulant apporter leur collaboration ne peuvent dès lors que soumettre leurs propositions de modification à ces responsables enregistrés, qui décideront d'inclure cette contribution ou non.

¹¹⁹ Il faut noter que des alternatives apparaissent : le programme Subversion par exemple. Voir <http://subversion.tigris.org/>.

¹²⁰ Ce système d'extraction, mémorisation et fusion à la demande des modifications apportées est basée sur les commandes « diff » et « patch ».

Cette possibilité de récupérer toute version est parfois laissée au « public ». Parfois au contraire, le « public » n'a accès qu'aux versions officielles¹²¹.

B. Titularité des droits d'auteur sur un logiciel libre

1) Impossibilité d'une qualification globale

85. Au vu de ce qui précède, se pose donc la question de savoir de quelle façon on peut qualifier un logiciel libre. La tentation est grande d'appréhender le logiciel libre dans sa globalité évolutive. Les uns y verront une œuvre de collaboration¹²², d'autres une œuvre collective¹²³. Dans ce genre d'exercice de qualification unique, les notions d'œuvres primaires et dérivées sont bien entendu inappropriées.

86. La qualification d'œuvre collective est séduisante, dans la mesure où elle permettrait une centralisation des droits d'auteur dans le chef du seul entrepreneur, ou initiateur de l'œuvre.

D'emblée, l'absence du concept d'œuvre collective en droit belge nous oblige à fermer cette voie. Cependant, cette qualification doit à notre avis être écartée pour plusieurs autres raisons majeures. Tout d'abord, les éléments constitutifs de l'œuvre collective sont loin d'être réunis. S'il est vrai qu'un logiciel libre est souvent issu de l'initiative d'un ou plusieurs développeurs principaux, l'ensemble des noms de tous les participants au développement des logiciels libres sont généralement repris dans un fichier « Authors » livré avec le programme, et le(s) développeur(s) principal (aux) ne prétendra (ont) en aucun cas être le(s) seul(s) titulaire(s) de droits d'auteur¹²⁴. Il est de la nature même de l'esprit du mouvement libre de voir dûment reconnue la participation de chacun au développement d'un programme. Ensuite, le développement d'un logiciel libre n'est pas toujours organisé, du début à la fin de son évolution, par les

¹²¹ Pour plus d'information sur le système CVS, voir M. BAR et K. FOGEL, *Open Source Development with CVS*, 3^{ème} Ed., Scottsdale Arizona, Paraglyph press inc., 2003, disponible à l'adresse http://cvsbook.red-bean.com/OSDevWithCVS_3E.pdf.

¹²² M. CLEMENT-FONTAINE, *La licence publique générale GNU*, mémoire de DEA Droit des Créations Immatérielles, année 1998-1999, <http://www.juriscom.net/uni/mem/08/log01.pdf>.

¹²³ J-B. SOUFRON, « La Licence Publique Générale : un système original de protection juridique pour les créations issues des systèmes de développement coopératifs », <http://soufron.free.fr/files/gpl.pdf>.

¹²⁴ L'un des principaux effets recherchés dans les différentes licences libres est justement de garantir à tous les auteurs ayant successivement contribué à la création du logiciel leur droit de paternité. G. AZZARIA, « Les logiciels libres à l'assaut du droit d'auteur », *Les Cahiers de Propriété Intellectuelle*, janvier 2004, vol. 16, n°2, p. 416.

mêmes développeurs principaux. Le logiciel peut subir plusieurs évolutions différentes, parallèles et parfois même concurrentes qui prendront la forme de « forks »¹²⁵, chacune dirigée par des développeurs principaux différents. De même, toute personne étant en possession des codes sources du programme diffusé sous une licence libre peut le réutiliser ou le développer à sa guise, sans tenir compte de quelconques directives de coordinateurs de projets.

87. L'œuvre de collaboration pourrait à première vue constituer la meilleure « représentation légale » de l'esprit du Libre. Il est en effet tentant de percevoir le logiciel libre comme étant cette grande œuvre de collaboration perpétuellement inachevée et en constante évolution, édifice auquel chacun apporte sa pierre dans un esprit communautaire, se voyant au passage reconnaître le statut d'auteur de l'œuvre. Cependant, cette analyse doit également être écartée, pour diverses raisons dont certaines déjà exposées ci-dessus. Tout d'abord, au point de vue des éléments constitutifs de l'œuvre de collaboration, l'on doit se rendre à l'évidence que la situation de concours disparaît à un moment ou à un autre. Il est clair qu'au moment du lancement d'un projet les développeurs du programme se connaissent et ont la volonté de coopérer afin de créer une œuvre commune. Mais la première mise à disposition du public du code du programme en question aura pour conséquence que quiconque pourra faire ce qu'il veut de ce code. Si certains programmeurs prendront contact avec les responsables du projet et participeront à la collaboration, d'autres le développeront séparément ou y puiseront ce qui les intéresse, sans plus. Par ailleurs, la constitution de « forks » nous paraît également s'opposer à la qualification d'œuvre globale de collaboration. Enfin, si ces arguments d'ordre factuel se voient contestés sur la base d'une présentation du logiciel libre comme étant un immense projet commun, où la collaboration s'apparente à l'adhésion à un mouvement collectif global, cette adaptation du concept de « collaboration » ne résisterait de toute façon pas à une application correcte des règles de l'indivision mise en place par la loi. En effet, prétendre que le logiciel libre est une œuvre de collaboration globale, c'est accorder à tous les participants dont les apports peuvent être qualifiés d'originaux des droits identiques d'auteur sur l'œuvre de collaboration. Par application des articles 4 et 5 de la LDA, cela impliquerait que tous ces collaborateurs doivent se concerter afin d'exercer les droits d'auteur relatifs à l'œuvre d'un commun accord. Or, seuls les auteurs originaires, lors de la divulgation du programme (exercice du droit moral), choisissent la licence libre sous laquelle ce programme est distribué (exercice de droits patrimoniaux). Tous les « collaborateurs » postérieurs ne participent donc pas à cette décision. Par ailleurs, si la licence choisie est de type « copyleft », ces collaborateurs

¹²⁵ L'on entend par « fork » le fait qu'un groupe de participants à un projet décide de se désolidariser des autres participants et de développer davantage le programme « de leur côté ».

subséquents se voient imposer cette licence pour tout développement futur. On est donc bien loin de la définition légale de l'œuvre de collaboration.

2) Fractionnement temporel

88. Il faut dès lors se rendre à l'évidence et admettre que l'évolution d'un programme informatique n'est qu'une chaîne d'œuvres dérivées, dont l'analyse détaillée ne peut être correctement effectuée qu'en accomplissant un fractionnement temporel, comparable à une échelle dont chaque échelon équivaut à la création d'une nouvelle œuvre dérivée.

Afin d'obtenir un aperçu complet de la titularité des droits sur un logiciel libre tout le long de son évolution, il s'agit dès lors de repérer chaque nouvelle œuvre dérivée et de déterminer le(s) titulaire(s) des droits d'auteur sur cette dernière, en ne perdant jamais de vue que cette œuvre dérivée n'a été créée qu'avec l'autorisation du (des) titulaire(s) des droits sur l'(les) œuvre(s) dont elle reprend des éléments originaux. Cet exercice est loin d'être évident¹²⁶.

Pour entrer dans le détail de cette étude, nous ferons une distinction entre l'œuvre primaire et les œuvres dérivées subséquentes.

a) Œuvre primaire

89. Dans le cadre de cette analyse, l'œuvre primaire sera considérée comme étant la première version du programme que l'auteur aura divulguée¹²⁷ et sur laquelle ce dernier aura exercé un droit patrimonial (*infra*, n° 113 et s.). En terme de multiplicité d'auteurs, l'œuvre primaire constitue la première œuvre de collaboration « achevée »: une fois en présence d'une œuvre primaire, les modifications subséquentes apportées à l'œuvre donneront naissance à une œuvre dérivée.

¹²⁶ P. LAMBERT, « Copyleft, Copyright and Software IPRs: is Contract Still King? », E.I.P.R., 2001, p. 170 : « Who or how many people may retain copyright in Linux or some other large open-source software projects is a problem. Can each person have copyright of their own contribution or is it licenced/assigned to somebody else? What of those contributing new patches or developments to the overall program? Can rights exist in persons unknown, from places unknown, at times unknown? »

¹²⁷ Il n'est pas sûr que les programmeurs bénéficient du droit moral de divulgation. Voy. *infra*, n° 162 et s.). En tout état de fait, ceci n'entre pas en ligne de compte dans le développement présent ; en effet, nous nous servons des principes de la divulgation dans le seul but de déterminer à quel moment nous sommes en présence d'une œuvre primaire. Une fois ce fait constaté, peu importe, à ce stade, de savoir si le programmeur bénéficie ou pas d'un « droit de divulgation ».

90. Une partie de la doctrine s'accorde à présenter le droit de divulgation comme étant le droit de décider quand l'œuvre est achevée et quand elle est digne d'être présentée au public¹²⁸. Suivant ce raisonnement, toute oeuvre divulguée devrait être considérée comme étant achevée par l'auteur. D'autres auteurs ne reprennent pas la notion d'achèvement de l'œuvre, ne retenant que le fait de décider quand porter l'œuvre à la connaissance du public¹²⁹. A première vue, cette présentation correspond davantage à l'esprit du mouvement libre et à son idéal communautaire. Outre cette divergence doctrinale sur la définition du droit de divulgation, les contours de ce droit moral sont d'autant plus flous qu'ils présentent une dimension psychologique. En effet, la volonté de l'auteur est aussi déterminante que l'acte matériel de publication (qui peut par ailleurs s'effectuer sans cette dernière)¹³⁰. Par ailleurs, on pourrait imaginer que la communication d'ébauches de programme à la communauté représente une invitation à la collaboration et non une véritable « divulgation ». Dès lors, le concept de divulgation à lui seul n'est pas d'un grand secours pour déterminer quand l'œuvre primaire (et éventuellement de collaboration) se termine et quand commence l'élaboration d'une œuvre dérivée.

91. Cependant, lorsque le logiciel est communiqué au public accompagné d'une licence libre, ou, en d'autres termes, lorsque les droits patrimoniaux relatifs à ce logiciel sont exercés, le doute n'est plus permis. En matière de collaboration, ces droits sont indivis et exercés d'un commun accord par l'ensemble des collaborateurs¹³¹. Il n'est donc plus question de prétendre à la proposition de collaboration puisque le cercle de collaborateurs est déjà clos.

Nous considérerons dès lors être en présence d'une oeuvre primaire « achevée » lorsque ce logiciel est divulgué¹³² au public accompagné d'une licence, en insistant cependant sur le fait que le terme « achevée » n'implique pas que le logiciel soit complet, et encore moins opérationnel.

¹²⁸ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 82, n° 81 ; A. BERENBOOM, *op. cit.*, p. 132 ; F. POLLAUD-DULIAN, « Le droit moral en France », in *Le droit moral de l'auteur*, ALAI, congrès d'Anvers 19-24 septembre 1993, p. 133.

¹²⁹ Fernand DE VISSCHER et Benoît MICHAUX définissent le droit de divulgation comme étant : « le droit de l'auteur de décider seul si, quand et sous quelles conditions l'œuvre sera portée à la connaissance du public » (F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 151 et s.). André et Henri-Jacques Lucas présentent ce droit comme étant le droit de porter l'œuvre à la connaissance du public (A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 313 et s.).

¹³⁰ A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 313 et s.

¹³¹ P. WAUWERMANS, *Le droit des auteurs en Belgique*, Bruxelles, Société Belge de Librairie, 1894, n°115 ; A. BERENBOOM, *op. cit.*, 1997, n°109 ; A. PUTTEMANS, *op. cit.*, 1999, p. 358.

¹³² Il nous faut en effet conserver la notion de divulgation, dans la mesure où il est possible d'exercer des droits patrimoniaux sur l'œuvre, et à fortiori concéder des licences, avant même que l'œuvre ne soit créée. Dans ce cas, c'est la divulgation qui permettra l'usage effectif des droits concédés.

92. Une fois cette notion dégagée, la suite dépend des faits. Si le logiciel primaire est l'œuvre d'une seule personne, cette dernière sera l'auteur unique de l'œuvre et en détiendra tous les droits d'auteur. Si plusieurs personnes, agissant dans un esprit d'inspiration commune et apportant leur participation originale, ont collaboré à l'élaboration du programme, ce dernier sera une œuvre de collaboration¹³³.

93. L'auteur ou les auteurs ainsi désignés détiendront l'ensemble des droits d'auteur sur le programme primaire, ce qui a pour conséquence principale qu'ils décident de la licence sous laquelle le logiciel primaire sera distribué. Ce choix, portant entre autres sur le caractère « copyleft » ou non de la licence, aura une influence déterminante sur toute l'évolution du programme et sur la liberté des auteurs d'œuvres dérivées d'exercer leurs droits :

- S'ils optent pour une licence non-copyleft (par exemple, la B.S.D.), la liberté des auteurs d'œuvres dérivées sera entière : ils pourront déterminer le sort de ces dernières et les distribuer sous d'autres licences, libres ou propriétaires.
- A l'inverse, s'ils choisissent une licence copyleftée (la G.P.L. par exemple), la condition de leur concession de licence est la distribution des œuvres dérivées sous les mêmes conditions, à savoir, la même licence.

b) Œuvres dérivées successives

Première œuvre dérivée

94. Les programmeurs désirant modifier le programme, en intégrer tout ou partie dans un autre logiciel, redistribuer le programme... disposent de tous les éléments nécessaires : des codes sources et une licence leur permettant d'accomplir pareils actes. Avant de poser tout acte d'exploitation, ces programmeurs devront cependant prendre la peine de bien lire cette dernière, car selon les types de licences libres, leurs obligations, contre-parties de la licence, seront plus ou moins importantes.

¹³³ En France, la qualification d'œuvre collective pourra également être envisageable, à condition toutefois que l'œuvre primaire réponde aux conditions, ce qui nous paraît peu probable. Voy. n° 74 et s.

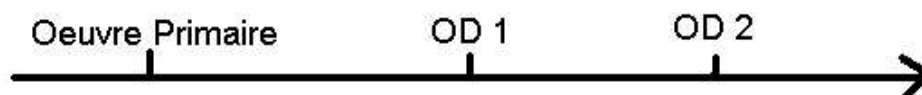
Si les modifications apportées au code consistent en un apport original, l'œuvre qui en résulte sera considérée comme dérivée¹³⁴. Si plusieurs personnes ont participé à sa conception, cette œuvre dérivée peut également consister en une œuvre de collaboration, à condition que les éléments constitutifs ci-dessus définis soient présents.

95. L'auteur du programme dérivé, ou les auteurs en cas d'indivision, sera (seront) titulaire(s) de tous les droits afférents à cette œuvre. Cependant, il faudra toujours tenir dûment compte des droits des auteurs de l'œuvre primaire. Le choix de la licence sous laquelle le logiciel dérivé sera distribué peut être imposé : c'est entre autres le cas des systèmes copyleft. Par ailleurs, même si cette licence permet l'utilisation, la modification du programme dérivé ou la reprise en tout ou en partie de ses composantes dans un autre logiciel, il ne faudra pas perdre de vue qu'en cas de reprise d'éléments originaux de l'œuvre primaire, l'autorisation du (des) auteur(s) de cette dernière est également requise¹³⁵.

Chaîne d'œuvres dérivées

96. Chaque maillon de la chaîne d'œuvres dérivées s'analysera de la même façon qu'exposée ci-dessus. Chaque œuvre dérivée sera accomplie grâce à l'autorisation accordée par les auteurs des œuvres dérivées et/ou des œuvres primaires dont elle incorpore des éléments originaux.

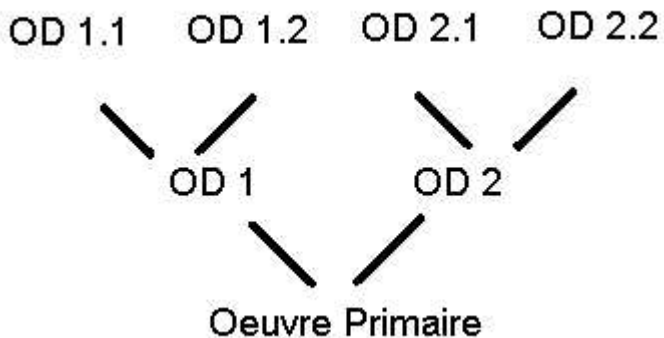
Le développement des logiciels peut s'avérer rapidement difficile à retracer. En effet, à des fins explicatives, nous présentons l'évolution du logiciel libre comme une chaîne d'œuvres dérivées, une sorte d'échelle dont chaque échelon représentait la création d'une nouvelle œuvre dérivée.



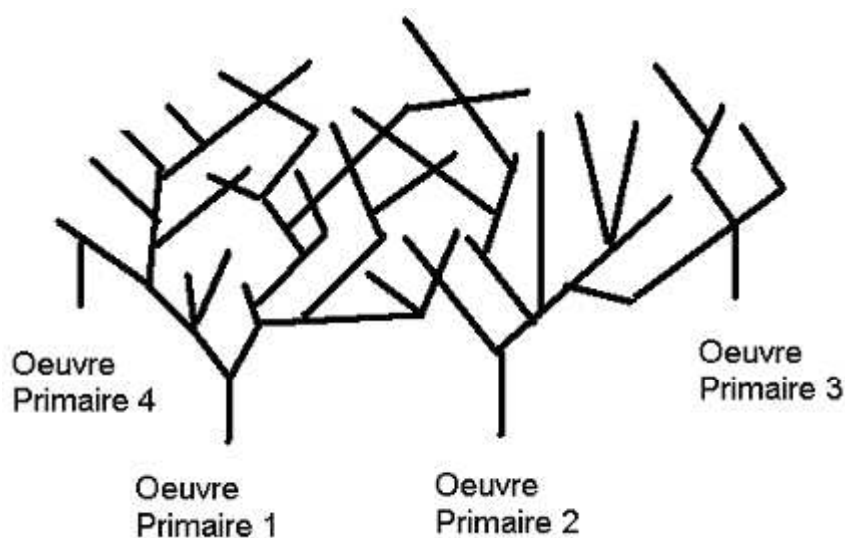
¹³⁴ Par contre, si les modifications apportées à l'œuvre primaire sont insuffisantes pour être considérées comme étant un apport original, le logiciel résultant n'est pas une œuvre dérivée, et le programmeur ne sera pas titulaire de droits d'auteur, ces derniers restant la propriété intellectuelle du (des) auteur(s) de l'œuvre primaire.

¹³⁵ Cette autorisation sera, bien entendu, systématiquement accordée, puisque cette œuvre primaire est distribuée sous licence libre. Nous aurons cependant l'occasion de voir dans la suite de nos analyses que les problèmes se poseront en cas de « retrait » du logiciel (cf. n° 104 et s.).

97. Cette présentation est cependant loin d'être complète. En effet, chaque œuvre peut être développée parallèlement par plusieurs programmeurs différents, et donner ainsi lieu à des œuvres dérivées différentes. L'évolution du logiciel ressemble dès lors davantage à un arbre qu'à une échelle...



98. Mais cette présentation n'est pas encore satisfaisante, car elle ne tient pas compte du fait qu'il est tout à fait envisageable de créer une œuvre dérivée sur la base de plusieurs œuvres antérieures différentes. L'on peut en effet très bien imaginer reprendre du code de plusieurs logiciels libres différents afin de créer une nouvelle application. Ainsi les logiciels libres primaires peuvent non seulement donner naissance à une évolution en arborescence, mais chaque branche peut également se voir ajouter une partie du code d'un logiciel primaire ou dérivé d'une arborescence parallèle. Nous représenterions dès lors le développement des logiciels libres comme une forêt dont les branches des arbres s'entrecroisent.



99. Qu'en est-il des logiciels développés par système CVS ? Quand nous décrivons le système CVS, nous faisons la distinction entre les participants « internes » au projet (les « responsables » du projet) et les personnes extérieures (*supra*, n° 83 et s.). Alors que les premiers ont leur propre « directory », permettant au système de tenir compte de leur travail à tout instant, les personnes extérieures n'ont pas la possibilité de travailler directement sur le projet, et doivent soumettre leurs propositions de modifications et de développements aux responsables.

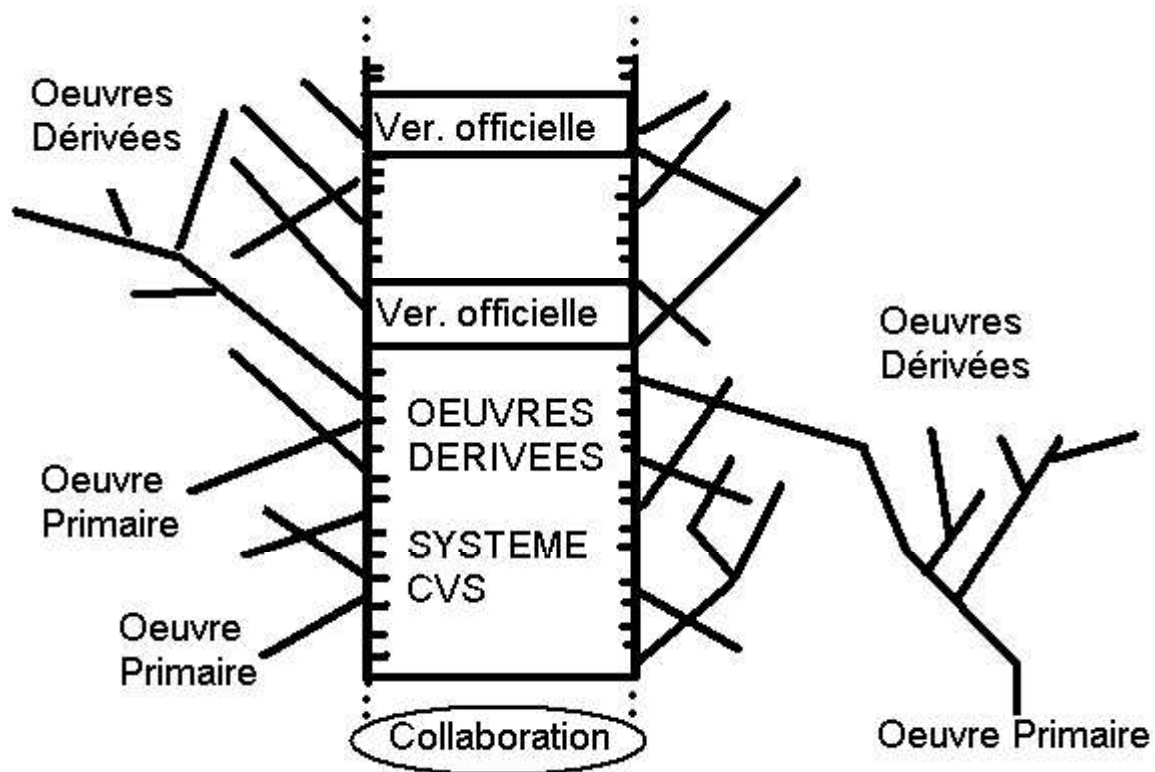
La situation entre les participants internes au projet est clairement qualifiable de collaboration. A l'inverse, les apports ponctuels de développeurs externes évoquent davantage la notion contestée d'apports de créateurs accessoires (*supra*, n° 59 et s.). Selon nous, ce genre d'apport s'analysera au cas par cas, eu égard aux relations entre les participants au projet et les personnes extérieures. Par ailleurs, il est fréquent que des personnes extérieures intègrent le groupe et se fassent attribuer un « directory »¹³⁶.

Lorsque nous parlons de collaboration au sein du système CVS, entendons-nous bien : ce système est un véritable générateur d'œuvres dérivées. Outre les versions officielles publiées quand les participants l'estiment opportun, le système peut fusionner à tout instant les apports effectués par les collaborateurs, avec possibilité de ne tenir compte des modifications apportées que jusqu'à un instant déterminé. Dès lors, il est possible de découper l'évolution du logiciel, modification par modification, en autant d'œuvres dérivées subséquentes, toutes cependant œuvres de collaboration issues généralement des mêmes auteurs, à savoir les participants au projet¹³⁷. Chacune de ces œuvres dérivées est soumise à la licence choisie pour le projet¹³⁸.

¹³⁶ Ce type d'apport devra être considéré comme une collaboration si son auteur a réellement fait partie intégrante de la coopération qu'appelle cette notion. L'attribution d'un directory à un nouvel arrivant correspondra indubitablement à son entrée dans le projet, et ce nouvel arrivant sera considéré comme un collaborateur au sens de la loi. A l'inverse, cet apport sera considéré comme étant une œuvre primaire reprise dans l'œuvre de collaboration, grâce à l'autorisation de l'auteur, si ce dernier ne s'est pas vu intégré au sein du groupe des collaborateurs.

¹³⁷ Sauf en cas d'inclusion de développeurs extérieurs au projet, ou s'il s'avère qu'un collaborateur externe a, pour un temps, participé au développement du programme en partageant l'« intimité spirituelle » des participants au projet. Dans ce cas, les œuvres dérivées doivent seulement être considérées comme étant issues d'une collaboration entre un nombre plus important de personnes.

¹³⁸ Il n'est également pas impensable que les participants au projet décident de changer de licence. Ce qui est envisageable, à condition que les termes des licences relatives à tous les éléments originaux présents dans le programme à ce moment soient compatibles avec les termes de cette nouvelle licence, ou à condition d'obtenir l'accord de tous les auteurs de tous les éléments originaux présents dans le programme à cet instant.



3) Risques du système

100. Nous avons pu constater ci-dessus que les logiciels libres étaient loin de suivre une évolution linéaire et indépendante. Au contraire, ils se présentent souvent sous la forme d'un patchwork de lignes de code provenant de tous horizons, aux auteurs multiples et variés, et issues de modifications successives.

Déterminer les titulaires de droits sur la dernière version d'un logiciel libre n'est pas difficile en soi : ce sont les auteurs qui ont créé cette dernière œuvre dérivée. Cependant, ils n'ont pu créer cette œuvre dérivée qu'en ayant l'autorisation préalable des auteurs de tous les éléments originaux intégrés dans l'œuvre, ce qui est normalement le cas puisque tous ces auteurs sont censés avoir chacun concédé ce droit aux auteurs subséquents par le biais de licences libres.

101. Dès lors, on remarque tout de suite les failles de ce système : il est basé sur la présomption que tous les auteurs de la chaîne ont tous respecté scrupuleusement les termes des licences et les règles du droit d'auteur.

102. Qu'en est-il en effet si l'un des auteurs de la chaîne n'a pas respecté les termes de la licence lui permettant de créer une œuvre dérivée et de la redistribuer ? L'œuvre résultante peut être qualifiée de contrefaçon et son auteur

poursuivi. Cependant cela ne devrait normalement pas affecter les auteurs subséquents de la chaîne, dans la mesure où ils bénéficient tous, malgré tout, des licences des auteurs précédant de la chaîne, y compris du contrefacteur en ce qui concerne les éléments originaux que ce dernier a apportés. Certaines licences prévoient explicitement cette solution¹³⁹.

103. Plus grave est le cas de figure où l'un des auteurs a intégré dans la chaîne, à l'insu des auteurs subséquents, des éléments originaux dont il n'est pas l'auteur et qui ne sont pas distribués sous licence libre¹⁴⁰. A ce moment, non seulement cet auteur se rend coupable de contrefaçon, mais il entraîne avec lui tous les auteurs subséquents (dans la mesure où les œuvres dérivées subséquentes intègrent toujours les éléments originaux en question)¹⁴¹.

104. On pourrait également s'interroger sur le cas où un des auteurs de la chaîne décide subitement de ne plus accorder de licence libre sur son œuvre... qu'il « retire » son programme du système libre¹⁴². Ce peut être le cas de l'auteur de l'œuvre primaire. Ce peut aussi être le cas de tous les auteurs d'œuvres dérivées, que le système de licence soit copyleft ou non¹⁴³. Ce faisant, l'auteur retient son autorisation à la création de toute œuvre dérivée future à laquelle seraient intégrés des éléments originaux dont il est l'auteur. A ce stade, il nous semble que les termes de la licence libre sous laquelle cet auteur distribuait son logiciel ont une influence capitale sur la solution à apporter à ce cas de figure. Si la licence permet à son bénéficiaire de sous-licencier¹⁴⁴ les

¹³⁹ Voir par exemple l'article 4 de la G.P.L. prévoyant : « *parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance* ».

¹⁴⁰ L'affaire SCO contre IBM, lancée le 6 mars 2004 par SCO et toujours actuellement pendante, est un bon exemple de ce genre de situation. SCO Group, Inc., société de développement informatique, a acquis les droits sur le programme UNIX, originellement développé par AT&T. SCO prétend avoir concédé des licences sur certains de ses logiciels à la société IBM. Les deux sociétés auraient également conclu un contrat de confidentialité concernant certains de ces programmes. SCO estimerait qu'IBM a violé les termes de ces contrats et outrepassé ses droits par sa contribution à l'élaboration du code de Linux, auquel IBM aurait inclus du code provenant des logiciels en question. Voir l'« amended complaint » de SCO déposée par SCO le 16 juin 2004, <http://www.sco.com/ibmlawsuit/amendedcomplaintjune16.html>; voir également L. ROSEN, *Open Source Licensing*, Upper Saddle River (New Jersey), Prentice Hall, 2004, p. 291.

¹⁴¹ En ce qui concerne l'existence d'éventuelles garanties d'éviction dans certaines licences, voy. *infra*, n° 297.

¹⁴² Que ce soit pour le distribuer sous licence privée ou non, à condition toutefois de respecter les licences relatives aux éléments originaux intégrés dans son œuvre dérivée.

¹⁴³ En effet, si le système copyleft impose que la distribution de l'œuvre dérivée se fasse sous la même licence libre, il n'oblige cependant pas la distribution.

¹⁴⁴ Sur les sous-licences en matière de licences libres, voir par exemple L. ROSEN, *op. cit.*, p. 87 et s.

droits concédés, la chaîne n'est pas rompue, et les futurs auteurs subséquents obtiendront l'autorisation nécessaire par le biais de sous-licences¹⁴⁵. A l'inverse, si la licence ne prévoit pas pareille clause, plus aucune œuvre dérivée subséquente ne pourra contenir d'éléments originaux pour lesquelles la licence libre aura été « retirée »¹⁴⁶.

Sous-section 3 : Transferts de droits

105. Jusqu'à présent, notre modèle général de titularité se basait sur une chaîne d'œuvres dérivées dans laquelle chaque auteur d'éléments originaux reste titulaire de son œuvre dérivée mais accorde des licences de type non-exclusif.

¹⁴⁵ Par exemple, la sous-licence est explicitement incluse dans les droits concédés par l'article 2.1, a) de la M.P.L., version 1.1. De même, la possibilité de sous-licencier est entre autres prévue aux articles 1 et 2 de l'O.S.L., version 2.1. Les licences Apache, version 2.0 (1^{er} janvier 2004)) et la MIT prévoient explicitement la sous-licence, alors que la B.S.D. n'en touche mot.

¹⁴⁶ A cet égard, les licences G.P.L., version 2 de Juin 1991 et L.G.P.L., version 2.1 de février 1999, méritent une attention spéciale. L'article 0, §2 de la G.P.L. prévoit : « *Activities other than copying, distribution and modification are not covered by this License; they are outside its scope* » (identique à l'article 0, §5 de la L.G.P.L.). L'on aurait tendance à penser que l'acte de sous-licencier n'est pas permis. Par contre, on peut lire à son article 4 : « *You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance* » (similaire à l'article 8 de la L.G.P.L.). L'on en déduit dès lors que la G.P.L. et la L.G.P.L. contiennent des dispositions relatives à la sous-licence. Sans doute est-ce le cas à l'article 2, b) : « *You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License* » (similaire à l'article 3, c) de la L.G.P.L.). Lawrence Rosen semble ne pas douter du fait que la G.P.L. autorise la sous-licence (L. ROSEN, *op. cit.*, p. 112). Il faut cependant ajouter que l'article 6, §1 de la G.P.L. prévoit : « *Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions* » (similaire à l'article 10 L.G.P.L.). On peut alors se demander de qui le licencié obtient la licence sur les éléments originaux du programme en question dont « son » donneur de licence n'est pas l'auteur : est-ce de son co-contractant (« son » donneur de licence lui octroyant une sous-licence), des auteurs de chacun de ses éléments (licence des auteurs originaux) ou est-ce de chacune de ses personnes? Nous pencherons pour cette dernière solution. Dans la mesure où les termes de la licence, quel que soit son concédant, sont identiques, cette solution semble ne constituer qu'un avantage supplémentaire pour le licencié. En ce qui concerne le système de stipulation pour autrui en cascade tel que prévu, par exemple, à l'article 10 de la G.P.L., voy. *infra*, n° 250 et s.

Cependant, il arrive fréquemment que la création, le développement ou l'amélioration d'un logiciel se fasse dans le cadre d'un contrat de travail ou de commande¹⁴⁷.

Les informaticiens employés sont en effet de plus en plus amenés à travailler sur la base de code libre. Or, selon l'article 3 de la LPO, il y a, sauf clause contraire, présomption de transfert¹⁴⁸ des droits patrimoniaux de l'employé à l'employeur.

Pareille présomption n'est pas prévue en matière de contrat de commande, mais la cession des droits est soumise dans certains cas à des conditions de forme assouplies (art. 3 LDA). Qui plus est, le modèle économique du logiciel libre est précisément basé sur la fourniture de services comprenant entre autres des tâches d'adaptation, modification, maintenance et autres développements sur mesure. Les contrats de commande seront dès lors courants, s'agissant de mettre au point des infrastructures informatiques « libres », et contiendront fréquemment des clauses de transfert.

Par ailleurs, il pourrait également être envisageable qu'en dehors de tout cadre de contrat de travail ou commande, un ou plusieurs auteurs cèdent leurs droits sur leur œuvre.

Nous aurons l'occasion de remarquer que le transfert de droits d'auteur n'influence pas le schéma de chaînes d'œuvres dérivées ci-dessus défini.

§ 1. Respect de la règle de l'auteur originaire – personne physique

106. Comme déjà analysé auparavant, l'article 6, alinéa 1^{er} de la LDA prévoit que c'est la personne physique qui a créé l'œuvre qui doit être considérée comme étant titulaire originaire du droit d'auteur, et cette règle ne souffre, en droit belge, aucune exception.

¹⁴⁷ Pour une analyse approfondie des règles en matière de contrat relatif aux droits d'auteur, voir *infra*, n° 181 et s.

¹⁴⁸ Nous ne nous étendons pas, dans le cadre de cette analyse, sur la notion de « cession » de droit, utilisé dans la loi, et qui est source d'ambiguïté dès lors qu'elle concerne aussi bien les « licences » que les « transferts » de droits. Nous renvoyons à ce sujet à la doctrine classique. Voir par exemple A. STROWEL et E. DERCLAYE, *op. cit.*, p. 104 ; A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 391 et s. Pour éviter tout malentendu, nous parlerons dès lors de transfert de droits d'un cédant à un cessionnaire, quand le premier « perd » toute prérogative patrimoniale au bénéfice du second qui s'en retrouve investi. A l'inverse, dans le cadre d'un contrat de licence, le donneur de licence conserve ses droits patrimoniaux, mais donne au licencié l'autorisation de poser certains actes sur l'œuvre en question.

Dès lors, tout employeur, cessionnaire par contrat de commande ou bénéficiaire d'un contrat de transfert ne pourra jamais prétendre être la première personne investie de droits d'auteurs, mais restera toujours cessionnaire de ceux-ci.

L'analyse de transfert de droits d'auteur sur un logiciel libre peut être subdivisée en deux parties. Le transfert des droits sur une œuvre primaire d'une part et le transfert des droits sur une œuvre dérivée d'autre part.

§ 2. Transferts de droits relatifs à une œuvre primaire

107. Cette première situation concerne le cas où le programmeur a conçu lui-même la totalité du logiciel.

108. S'agissant d'un travail d'employé ou d'un prestataire d'un contrat de commande (prévoyant le transfert des droits sur le logiciel en question), le cas d'espèce ne pose pas de problème : le transfert des droits s'effectuant, en vertu du contrat, à leur naissance, c'est le cessionnaire qui décidera du sort des droits patrimoniaux sur l'œuvre¹⁴⁹. Ainsi, une société pourra très bien, par exemple, décider de distribuer sous licence libre un programme conçu par ses employés.

109. Par contre, en ce qui concerne le contrat de transfert, l'on pourrait imaginer que l'auteur décide de distribuer son programme sous licence libre avant de transférer ses droits par la suite. Ce transfert ne posera aucun problème si le cessionnaire respecte ce choix de licence et continue à distribuer les programmes sous les mêmes conditions, ce qui peut d'ailleurs faire l'objet d'une clause explicite dans le contrat de transfert. Cependant, l'on ne voit pas quelle serait l'utilité de pareil transfert, et dès lors, il y a fort à parier que le cessionnaire aura acquis les droits dans le but de les exploiter de façon « propriétaire ». Le problème sera semblable à la situation de retrait telle qu'abordée précédemment, dans la mesure où les créateurs d'œuvres dérivées futures ne bénéficieront plus de l'autorisation nécessaire de la part du cessionnaire. Cependant, l'on peut également se demander si le cessionnaire doit respecter les contrats de licence libre passés antérieurement au transfert au cas où le contrat de transfert ne le prévoirait pas. Derrière cette interrogation se cache bien sûr l'épineux problème de la nature exacte des droits intellectuels : à quoi peut-on assimiler les droits conférés à un licencié ? A des droits réels ou personnels¹⁵⁰ ? Rares sont les auteurs qui se prononcent sur cette distinction, et

¹⁴⁹ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 220.

¹⁵⁰ Les droits réels sont des droits qui s'exercent directement sur les choses auxquelles ils ont trait, sans intermédiaire ou intervention d'autres sujets de droits (par exemple, le droit de propriété est un droit réel). A l'inverse, les droits personnels sont des liens juridiques entre plusieurs personnes. Par exemple, dans le cadre d'une location, le locataire ne bénéficie de la

les plus téméraires concluent que cette dichotomie n'a pas lieu d'être sur le terrain de la propriété intellectuelle¹⁵¹. Il nous paraît cependant impossible, sans nous baser sur cette *summa divisio*, de trancher la question qui nous préoccupe, à savoir s'il faut considérer que les droits du licencié sont comparables à des droits réels, qui lui octroieraient un droit « immédiat » sur l'œuvre, limité cependant aux actes autorisés par la licence¹⁵² ou si, au contraire, il ne peut poser ces actes que par l'intervention du titulaire, auquel il est juridiquement lié. Cette question fondamentale, étrangement peu abordée par la doctrine malgré ses conséquences multiples, ne peut être approfondie dans le cadre du présent ouvrage. Nous noterons simplement qu'Hendrik Vanhees affirme que les licenciés ne bénéficient que d'un droit personnel vis-à-vis du donneur de licence et non d'un droit réel sur le droit d'auteur¹⁵³. Suivant cette opinion, nous en arriverions à la conclusion que, sauf prévision de clauses spécifiques prévoyant le respect par le cessionnaire de toutes les licences antérieures concédées, l'auteur de l'œuvre primaire ne devrait pas conclure pareil contrat de transfert s'il a auparavant distribué son programme sous licence libre, au risque d'engager sa responsabilité contractuelle vis-à-vis des licenciés. En définitive, et tout bien considéré, ce cas de transfert ne devrait pas se rencontrer, sauf erreur ou mauvaise estimation de la part des parties.

§ 3. Transferts de droits relatifs à une œuvre dérivée

110. Il arrivera fréquemment qu'une entreprise demande à ses employés ou à des informaticiens externes de modifier un logiciel libre afin de l'adapter à ses besoins spécifiques. L'on pourrait aussi théoriquement imaginer un contrat indépendant de transfert de droits sur pareil programme.

La règle générale s'applique en l'espèce : l'on ne peut transférer plus de droits qu'on en a. Dans la mesure où l'exploitation d'une œuvre dérivée nécessite l'autorisation des titulaires de droits sur l'œuvre originale, le cessionnaire des droits sur le programme dérivé devra respecter la licence libre sous laquelle le logiciel original est distribué.

chose louée qu'en vertu des liens juridiques qui le lient avec le propriétaire de celle-ci ; le locataire n'a aucun droit direct sur cette chose. Voir par exemple H. DE PAGE, *Traité élémentaire de droit civil belge*, t. I, Bruxelles, Bruylant, 1962, p. 189 et s.

¹⁵¹ A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 392.

¹⁵² ...et ce, peu importe l'identité du titulaire des droits d'auteur dans le futur.

¹⁵³ H. VANHEES, *Een juridische analyse van grondslagen, inhoud en draagwijdte van auteursrechtelijke exploitatiecontracten*, Anvers, Maklu, 1993, p. 119 et s.

Sous-section 4 : Présomption de titularité de l'article 6, al. 2 de la LDA

111. « Est présumé auteur, sauf preuve contraire, quiconque apparaît comme tel sur l'oeuvre, du fait de la mention de son nom ou d'un sigle permettant de l'identifier ». Cette présomption simple de titularité joue aussi bien en faveur des auteurs originaires que des cessionnaires des droits sur l'œuvre. Le terme « auteur » doit en effet s'interpréter comme désignant tout titulaire de droits d'auteur, y compris toute personne morale, titulaire dérivé de ces droits. La présomption ne vaut qu'à l'égard des tiers (contrefacteurs, usurpateurs, utilisateurs, licenciés, etc.) mais non vis-à-vis des véritables auteurs¹⁵⁴. Ce système a en effet pour objectif de faciliter la tâche des victimes de contrefaçons, en mettant la preuve à charge des contrefacteurs¹⁵⁵.

112. Cette présomption s'avère souvent fort utile pour les auteurs d'œuvres traditionnelles. Cette règle est également en vigueur en matière de programmes informatiques¹⁵⁶. Son application aux logiciels libres laisse cependant perplexe, spécialement au vu des clauses de certaines licences imposant le respect des notices de copyright ou d'autres mentions.

Nous avons vu que les titulaires de logiciels libres, œuvres dérivées, sont uniquement les auteurs qui ont créé cette œuvre dérivée. Si leur autorisation est nécessaire à la modification et à la redistribution du logiciel, les auteurs antérieurs ne sont cependant pas à considérer comme auteurs de cette dernière version.

Néanmoins, certaines licences (la B.S.D. par exemple) prévoient l'obligation de respecter diverses notices de copyright, y compris leur apposition sur les œuvres dérivées. Le nom de l'auteur originaire apparaît dès lors sur l'œuvre dérivée, alors même qu'il n'a pas collaboré à sa création. Par contre, d'autres licences (c'est le cas de la M.P.L.) détaillent plus en profondeur le contenu des notices à apposer, prévoyant entre autres une distinction entre l'auteur originaire et l'auteur de l'œuvre dérivée et imposant de fournir une documentation des modifications apportées par ce dernier. Par ailleurs, un standard GNU impose que les logiciels libres contiennent toujours un fichier (« AUTHORS »)

¹⁵⁴ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 40 et s. ; F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 46 et s.

¹⁵⁵ J.-L. GOUTAL, « Présomption de titularité des droits d'exploitation au profit des personnes morales : la Cour de cassation maintient sa jurisprudence », *R.I.D.A.*, janvier 1998, p. 65.

¹⁵⁶ Bruxelles (9^{ème} Ch.), 11 avril 1997, *A. & M.*, 1997, p. 265, note V. VANOVERMEIRE ; Civ. Verviers (sais.), 3 avril 1998, *I.R.D.I.*, 1998, p. 230 ; Civ. Bruxelles (sais.), 3 oct. 2000, *I.R.D.I.*, 2001, p. 32.

De même, la disposition équivalente en droit français, à savoir l'article L. 113-1, s'applique également aux logiciels. Voir A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 163.

reprenant l'ensemble des personnes qui ont participé au projet. Il faut d'emblée préciser qu'en pratique, ces listes sont souvent très extensives et comportent également, par exemple, les noms des testeurs. Figurent donc sur ces listes non seulement des personnes qui n'ont pas participé à la création de l'œuvre dérivée à laquelle est jointe ce fichier, mais également des intervenants qui ne bénéficieraient normalement pas de droits d'auteurs.

Une attention particulière devra être portée au système de licence utilisée tout au long de l'évolution du logiciel, car il sera, selon nous, parfois possible d'y trouver les preuves contraires permettant d'écarter la présomption de l'article 6, § 2.

CHAPITRE 2 : LES DROITS PATRIMONIAUX ET LEUR EXERCICE

113. Les titulaires de droits d'auteur sur un logiciel bénéficient de prérogatives de deux types : patrimonial et moral. Nous ne nous pencherons, dans le cadre de la présente contribution, que sur les droits patrimoniaux, renvoyant pour la seconde catégorie à la contribution de Fabrice de Patoul au présent ouvrage.

Les droits patrimoniaux reconnus à l'auteur sont des droits exclusifs de poser certains actes sur l'œuvre, de les autoriser ou de les interdire. Ces prérogatives exclusives étaient au départ reconnues aux auteurs afin de permettre à ces derniers de monnayer leur autorisation et, au final, de vivre de leur art. Nous avons déjà pu remarquer qu'en matière de logiciels libres, cette autorisation était accordée en échange non pas d'argent mais du respect de certaines obligations.

Au sein de cette section, nous ferons d'abord une présentation des droits patrimoniaux classiques. Par la suite, nous aborderons brièvement la question du droit de destination, dont l'existence est contestée en droit d'auteur classique, et qui selon nous trouve encore moins de raison d'être en matière informatique. Nous passerons ensuite aux exceptions prévues par la LPO. Enfin, nous terminerons par le mécanisme des licences libres.

Section 1 : Les droits patrimoniaux classiques

114. Les auteurs se voient reconnaître deux prérogatives patrimoniales principales : le droit de communication au public et le droit de reproduction (comprenant les droits d'adaptation, de traduction, de location et de prêt) (art. 1,

§ 1 LDA). La directive « droit d'auteur et société de l'information »¹⁵⁷ en son article 4 reconnaît une troisième prérogative que la doctrine traditionnelle estime englobée dans celle de la reproduction¹⁵⁸ : le droit de distribution.

115. En ce qui concerne les droits patrimoniaux relatifs aux programmes informatiques, la LPO reprend dans son article 5 la liste d'actes exclusifs telle qu'établie dans la Directive sur la protection juridique des programmes d'ordinateur :

« Sous réserve des articles 6 et 7, les droits patrimoniaux comprennent :

- a) la reproduction permanente ou provisoire d'un programme d'ordinateur, en tout ou en partie, par quelque moyen et sous quelque forme que ce soit. Lorsque le chargement, l'affichage, le passage, la transmission ou le stockage d'un programme d'ordinateur nécessitent une telle reproduction du programme, ces actes seront soumis à l'autorisation du titulaire du droit;
- b) la traduction, l'adaptation, l'arrangement et toute autre transformation d'un programme d'ordinateur et la reproduction du programme en résultant sans préjudice des droits de la personne qui transforme le programme;
- c) toute forme de distribution au public, y compris la location et le prêt, de l'original ou de copies d'un programme d'ordinateur. La première vente d'une copie d'un programme d'ordinateur par le titulaire du droit ou avec son consentement épuise le droit de distribution de cette copie dans l'Union européenne, à l'exception du droit de contrôler des locations et des prêts ultérieurs du programme d'ordinateur ou d'une copie de celui-ci. »

On constate que, même si l'agencement des différents droits de cette liste diverge de la conception synthétique des droits, on y retrouve cependant les composants du droit classique de reproduction. Le droit de distribution y est explicitement cité.

116. Le droit de reproduction est certainement le droit le plus important en matière informatique puisqu'il permet aux auteurs un contrôle entier sur l'usage de leurs oeuvres. En effet, en incluant explicitement la reproduction provisoire au sein des actes de reproduction nécessitant l'autorisation de l'auteur, et eu

¹⁵⁷ Directive 2001/29/CE du Parlement et du Conseil 22 mai 2001 sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information, *J.O.C.E.*, 22 juin 2001, n° L 167/10.

¹⁵⁸ Selon la théorie classique de la conception synthétique des droits. Voir A. BERENBOOM, *Le nouveau Droit d'Auteur et les droits voisins*, Bruxelles, Larcier, 1997, n° 70, p. 105 ; A. STROWEL et E. DERCLAYE, *op. cit.*, p. 57, n° 51.

égard au fait que toute utilisation d'un logiciel implique, presque systématiquement¹⁵⁹, reproduction de tout ou partie de son code dans une mémoire temporaire (RAM, Cache,...) de l'ordinateur qui l'exécute, le législateur a dès lors reconnu aux titulaires des droits la possibilité de soumettre à leurs conditions chaque utilisation du logiciel en question¹⁶⁰. Cette prérogative deviendrait vite exorbitante si elle n'était pas contre-balancée par l'exception d'usage de l'article 6, §1, que nous aurons l'occasion d'analyser par la suite.

Le droit de reproduction porte aussi bien sur la reproduction à l'identique que la reproduction intellectuelle, c'est-à-dire l'emprunt de ce qui constitue l'originalité de tout ou partie de l'œuvre¹⁶¹. Dès lors, la traduction d'un programme dans un autre langage sera considérée comme étant un acte couvert par le droit de reproduction de l'auteur et nécessitant son autorisation.

117. Le droit de communication au public n'est pas repris dans cette liste. Aucun doute n'est cependant permis. La doctrine estime, d'une part, que cette liste n'est pas limitative, d'autre part, que l'assimilation des logiciels aux œuvres littéraires implique que les droits patrimoniaux reconnus à leurs auteurs comprennent également le droit de communication au public, malgré son absence dans la LPO¹⁶². Nous nous rallions à ce point de vue, dans la mesure où la reconnaissance aux auteurs de logiciels du droit de communication au public semble correspondre aux prescriptions du traité de l'OMPI du 20 décembre 1996 sur le droit d'auteur¹⁶³. La communication au public, en matière informatique, consistera entre autres en la diffusion (mise à disposition) d'un programme sur Internet.

118. Inutile de préciser que tous ces actes nécessitant l'autorisation de l'auteur concernent aussi bien les codes sources, objets ou exécutables, tous étant protégés par le droit d'auteur.

¹⁵⁹ Les cas d'exécution de code sans reproduction existent cependant. Tel est le cas de certains codes embarqués. Voy. *infra*, n° 152 et s.

¹⁶⁰ T. TJONG TJIN TAL, « De omvang van het gebruiksrecht van programmatuur », *Computerrecht*, 2001/2, p. 60.

¹⁶¹ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 65.

¹⁶² A. STROWEL et E. DERCLAYE, *op. cit.*, p. 215, n° 243; F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 221, n° 259 ; F. BRISON et J.-P. TRIAILLE, *op. cit.*, p. 143.

¹⁶³ Alors que l'article 4 de ce traité confirme que « les programmes d'ordinateur sont protégés en tant qu'œuvres littéraires au sens de l'article 2 de la Convention de Berne » et que « la protection prévue s'applique aux programmes d'ordinateur quel qu'en soit le mode ou la forme d'expression », son article 8 précise clairement que « les auteurs d'œuvres littéraires et artistiques jouissent du droit exclusif d'autoriser toute communication au public de leurs œuvres par fil ou sans fil, y compris la mise à la disposition du public de leurs œuvres de manière que chacun puisse y avoir accès de l'endroit et au moment qu'il choisit de manière individualisée ».

Section 2 : Droit de destination

119. Afin de terminer le tour d'horizon des droits patrimoniaux des auteurs, nous aventurer sur le terrain du droit de destination est un mal nécessaire, même si, ce sol est non seulement meuble, mais également un des plus vastes champs de bataille doctrinale. Cette théorie compte en effet ses partisans¹⁶⁴ et détracteurs¹⁶⁵.

120. Frank Gotzen, le premier auteur à affirmer l'existence de ce droit¹⁶⁶, définit le droit de destination comme étant « la faculté exclusive que possède l'auteur de réserver à un usage bien déterminé les reproductions de son œuvre qui ont été mises dans le commerce »¹⁶⁷.

L'on présente généralement l'arrêt de la Cour de cassation du 19 janvier 1956¹⁶⁸ comme étant la première application du droit de destination. Dans cet arrêt, la Cour estime que lorsqu'un titulaire de droit permet la reproduction d'une œuvre musicale sur un support pour usage privé, ce support ne peut être utilisé pour diffuser l'œuvre sur les ondes, même si le droit de communication au public a préalablement été octroyé. Les cours et tribunaux ont, depuis lors, appliqué à quelques rares reprises ce raisonnement, jugeant par exemple que le titulaire des droits d'auteur sur des planches de bande dessinées, tout en ayant autorisé la reproduction de celles-ci sur chaque page d'un calendrier, peut s'opposer à ce que l'on sépare ces planches du calendrier pour les vendre individuellement dans des cadres¹⁶⁹.

En résumé, ce droit serait un démembrement du droit de reproduction, et consisterait plus particulièrement en la soumission de l'autorisation de reproduire l'œuvre à la condition que cette reproduction ne soit utilisée que d'une certaine façon. Ce droit aurait dès lors pour effet de permettre à l'auteur un contrôle sur l'usage qui est fait des exemplaires de son œuvre.

¹⁶⁴ Voir par exemple F. GOTZEN, *Het bestemmingsrecht van de auteur. Onderzoek van de bevoegdheid van de auteur om derden een bepaald gebruik van in de handel gebrachte reproducties van zijn werk te verbieden, met inachtneming van rechtsvergelijkende en kartelrechtelijke gegevens*, Bruxelles, Larcier, 1975.

¹⁶⁵ A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 212 et s. ; S. DUSOLLIER, *Pour une protection appropriée des mesures techniques en droit d'auteur*, Thèse F.U.N.D.P., Namur, 12 mai 2004, Tome II, p. 360 et s.

¹⁶⁶ F. GOTZEN, *op cit.*

¹⁶⁷ F. GOTZEN, « Le droit de destination en Europe », *Dd'A*, 1989, p. 230.

¹⁶⁸ Cass. (1^{ère} Ch.), 19 janvier 1956, *R.C.J.B.*, 1956, p. 172, et note de J.-G. RENAULD, « Les destinées récentes du droit de reproduction mécanique et le droit d'auteur », *Ing. Cons.*, 1956, p. 65.

¹⁶⁹ Civ. Bruxelles, 15 février 1996, *A&M*, 1996, p. 319, note L. VAN BUNNEN.

121. En matière informatique, bon nombre d’auteurs s’accordent à dire que certaines dispositions de la directive, qui furent reprises dans la LPO, s’opposent à la reconnaissance du droit de destination. Tout d’abord, nous avons souligné que la LPO ne respectait pas l’agencement et le caractère synthétique de la LDA, en reconnaissant explicitement le droit de distribution. Nous rejoignons dès lors à Alain Strowel et Estelle Derclaye qui estiment peu compatible de faire coexister le droit de destination et le droit de distribution (légalement consacré et limité par la règle de l’épuisement)¹⁷⁰. La consécration de l’épuisement des droits constitue précisément un deuxième argument contraire à l’existence d’un droit de destination. En l’espèce, l’article 5, c de la LPO prévoit que « *la première vente d’une copie d’un programme d’ordinateur par le titulaire du droit ou avec son consentement épuise le droit de distribution de cette copie dans l’Union européenne, à l’exception du droit de contrôler des locations et des prêts ultérieurs du programme d’ordinateur ou d’une copie de celui-ci* ». Certains auteurs voient en pareille disposition une logique contraire à celle du droit de destination¹⁷¹.

Certains auteurs estiment qu’une reconnaissance du droit de destination aux auteurs d’œuvres informatiques découlerait de l’article 6, § 1 de la LPO, qui permet à la personne ayant le droit d’utiliser un logiciel d’accomplir les actes nécessaires à une utilisation conforme à sa « destination »¹⁷². Déduire l’existence d’un droit de destination d’une restriction à une exception au droit principal de reproduction, dont ce soi-disant droit est sensé être un démembrement, nous paraît être un raisonnement scabreux... Ainsi le droit de destination découlerait du retrait du droit de reproduction de certaines possibilités de contrôle ? Par ailleurs, et nous aurons l’occasion de revenir sur ce point lors de l’analyse des exceptions, nous verrons qu’une partie non négligeable de la doctrine, à laquelle nous nous rallions, n’accorde à ce critère de « destination du programme » qu’un caractère purement objectif (*infra*, n° 129).

122. Sans entrer davantage dans le débat sur l’existence ou la non-existence de cette prérogative en droit d’auteur classique, renvoyant à cet égard à la doctrine y afférente, quel serait le sens de reconnaître un droit de destination en matière de logiciels et quel intérêt pourraient y trouver les titulaires de droits sur les logiciels libres ?

¹⁷⁰ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 214. ; voir également A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 215.

¹⁷¹ A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 215.

¹⁷² P. KAMINA, *L’utilisation finale en propriété intellectuelle*, thèse, université de Poitier, 1996, n° 348 ; T. TJONG TJIN TAI, *op. cit.*, p. 61.

Ce droit est censé donner aux auteurs d'œuvres informatiques une plus grande maîtrise sur celles-ci, en leur permettant de restreindre l'usage qui est fait de chaque exemplaire matériel de ces dernières. Or nous avons vu que l'article 5, a) reconnaissait aux programmeurs le droit d'autoriser ou d'interdire toute reproduction, y compris toute reproduction provisoire. Dans la mesure où chaque utilisation de tout programme informatique entraîne reproduction en mémoire temporaire, chaque usage du programme est dès lors théoriquement soumis à l'autorisation de son auteur. C'est la principale raison pour laquelle les programmes informatiques font l'objet de licences d'utilisation. A première vue, le droit de reproduction pur et simple permet un contrôle sur l'œuvre informatique qui dépasse celui qui serait octroyé par la reconnaissance d'un droit de destination sur les exemplaires matériels¹⁷³.

123. On pourrait prétendre qu'un intérêt subsisterait malgré cet argument si l'on acceptait de donner au terme « exemplaire » une portée particulièrement large. En effet, en droit d'auteur traditionnel, on entend par exemplaire la copie physique de l'œuvre, à savoir un disque, un livre, un calendrier, ou toute autre reproduction graphique ou plastique. En définitive, il s'agit du « support » de l'œuvre. Or cette notion perd de son importance dans notre société de l'information, tant les données sont continuellement transférées d'un support à l'autre. Suivant cette optique, la notion d'exemplaire devrait également désigner le contenu du support¹⁷⁴, se dématérialiser afin de ne plus constituer que du code qui se transfère, s'efface et se copie au fil de son utilisation. Cette vision impliquerait que la copie éphémère du code effectuée dans les mémoires temporaires devrait également se voir qualifier d'« exemplaire » sur lequel s'exercerait un droit de destination¹⁷⁵. Cette nouvelle interprétation de la théorie du droit de destination semble cependant difficilement soutenable, car elle apparaît inconciliable avec l'exception d'usage de l'article 6, § 1^{er}.

¹⁷³ L'on peut cependant imaginer quelques situations théoriques de distribution où le droit de destination conserverait un certain intérêt. Par exemple, en cas de distribution d'exemplaires gratuits (CD-Roms), le droit de destination s'avèrerait utile afin d'empêcher une revente commerciale de ces exemplaires.

¹⁷⁴ ...tout en ne s'étendant pas, bien entendu, à l'œuvre elle-même et son immatérialité naturelle. Il faudrait dès lors considérer qu'entre le support « physique » et l'œuvre immatérielle, s'insère le code, contenu du support matériel et exemplaire immatériel de l'œuvre.

¹⁷⁵ Alors que la notion de droit de destination au sens traditionnel semble quasiment inutile aux auteurs de logiciels, une nouvelle interprétation extensive de ce droit donnerait aux titulaires de droits le pouvoir de limiter l'usage du code inscrit dans la mémoire vive, ce qui aurait été très utile pour légitimer la viralité des licences au copyleft « extrême » (*infra*, n° 135 et s.).

Section 3 : Exceptions aux droits patrimoniaux

124. En ses articles 6 et 7, la LPO reconnaît quatre exceptions aux droits patrimoniaux des auteurs de programmes d'ordinateurs. Certains auteurs présentent la règle d'épuisement prévue à la fin de l'article 5, et déjà abordée ci-dessus, comme étant une cinquième exception prévue par la LPO¹⁷⁶.

L'exhaustivité de cette liste est un sujet classique de controverses doctrinales. La question est de savoir si seules ces exceptions sont applicables en matière de programmes d'ordinateur ou si les exceptions prévues aux articles 21 et suivants de la LDA sont également de rigueur. Certains prétendent que la liste d'exceptions prévues dans la directive – et reprises dans la LPO – est exhaustive. Il serait dès lors exclu d'appliquer d'autres exceptions¹⁷⁷.

A la lumière du vingt-huitième considérant de la Directive¹⁷⁸, une doctrine majoritaire en Belgique considère que les exceptions de la LDA s'appliquent également en matière informatique, sauf lorsqu'une exception est déjà abordée par la LPO (à savoir principalement la copie privée)¹⁷⁹. Nous nous rallions à cette interprétation. Les exceptions de la LDA qui présentent un certain intérêt en ce qui concerne les logiciels seront entre autres la citation dans un but de critique, de polémique ou d'enseignement et la reproduction de courts fragments à l'occasion de comptes rendus des événements d'actualité. Cependant, vu l'aspect dérisoire de ces dernières, nous n'aborderons ci-dessous que les quatre exceptions prévues aux articles 6 et 7 de la LPO.

125. L'article 6, § 2, prévoit une exception impérative¹⁸⁰ permettant à la personne ayant le droit d'utiliser le programme d'ordinateur d'effectuer une copie de sauvegarde pour autant que cette copie soit nécessaire à l'utilisation du programme. Cette exception ne nécessite pas plus d'explication dans le cadre de cette contribution, étant donné que les bénéficiaires de licences libres se voient accorder d'office des droits bien plus étendus.

126. L'article 6, § 3, prévoit que « *la personne ayant le droit d'utiliser le programme d'ordinateur peut, sans l'autorisation du titulaire du droit, observer,*

¹⁷⁶ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 223 et s.

¹⁷⁷ B. CZARNOTA et R.J. HART, *Legal Protection of Computer Programs in Europe*, Londres, Butterworth, 1991, p. 31.

¹⁷⁸ « *Considérant que la présente directive n'affecte pas les dérogations prévues par les législations nationales, en application de la Convention de Berne, sur les points non couverts par la directive* ».

¹⁷⁹ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 223 et s. ; A. STROWEL et E. DERCLAYE, *op. cit.*, p. 243 ; F. BRISON et J.-P. TRIAILLE, *op. cit.*, p. 143.

¹⁸⁰ Voir l'article 8 de la LPO.

étudier ou tester le fonctionnement de ce programme afin de déterminer les idées et les principes qui sont à la base d'un élément du programme, lorsqu'elle effectue une opération de chargement, d'affichage, de passage, de transmission ou de stockage du programme d'ordinateur qu'elle est en droit d'effectuer ». Non sans un certain cynisme, un auteur a résumé cette exception impérative¹⁸¹ en disant qu'il est autorisé de regarder ce que l'on peut voir¹⁸². Le bénéficiaire d'une licence libre, qui aura accès au code source du programme, se passera bien entendu de cette exception dite d'observation.

127. L'article 7 prévoit l'exception de « décompilation »¹⁸³. Cette exception, également impérative¹⁸⁴, définit les cas dans lesquels il est permis, sans l'autorisation de l'auteur, de poser des actes de reproduction afin d'obtenir les informations nécessaires à la création d'un programme indépendant interopérable. Il s'agit entre autres de permettre, dans une certaine mesure, le *reverse engineering*. Cette disposition complexe et largement obscure a bien entendu fait couler pas mal d'encre. Nous n'entrerons pas non plus dans les détails, dans le cadre de notre étude, l'accessibilité aux codes sources des

¹⁸¹ Voir l'article 8 de la LPO.

¹⁸² M. VIVANT, « Logiciel 94 : tout un programme ? Loi n° 94-361 du 10 mai 1994 », *J.C.P.*, 1994, p. 434.

¹⁸³ L'article 7 de la LPO prévoit :

« § 1. L'autorisation du titulaire du droit n'est pas requise lorsque la reproduction du code ou la traduction de la forme de ce code au sens de l'article 5, a) et b) est indispensable pour obtenir les informations nécessaires à l'interopérabilité d'un programme d'ordinateur créé de façon indépendante avec d'autres programmes et sous réserve que les conditions suivantes soient réunies :

- a) les actes de reproduction et de traduction sont accomplis par une personne jouissant du droit d'utiliser une copie du programme, ou, pour son compte, par une personne habilitée à cette fin;
- b) les informations nécessaires à l'interopérabilité ne lui sont pas déjà facilement et rapidement accessibles;
- c) les actes de reproduction et de traduction sont limités aux parties du programme d'origine nécessaires à cette interopérabilité.

§ 2. Les dispositions du paragraphe précédent ne peuvent justifier que les informations obtenues en vertu de leur application :

- a) soient utilisées à d'autres fins que la réalisation de l'interopérabilité du programme créé de façon indépendante;
- b) soient communiquées à des tiers, sauf si ces communications s'avèrent nécessaires à l'interopérabilité du programme d'ordinateur créé de façon indépendante;
- c) ou soient utilisées pour la mise au point, la production ou la commercialisation d'un programme d'ordinateur dont l'expression est fondamentalement similaire, ou pour tout autre acte portant atteinte au droit d'auteur.

§ 3. Le présent article ne peut recevoir une application qui cause un préjudice injustifié aux intérêts légitimes du titulaire du droit, ou porte atteinte à l'exploitation normale du programme d'ordinateur. »

¹⁸⁴ Voir l'article 8 de la LPO.

logiciels libres épargnant à leurs utilisateurs toute peine en matière d'interopérabilité.

128. Reste enfin l'article 6, § 1^{er}. Le choix de présenter cette exception en dernier lieu n'est évidemment pas innocent. Malgré le fait qu'à première vue, cette exception est tout aussi inutile aux bénéficiaires de licences libres, puisque ceux-ci se voient octroyer des droits bien plus étendus, nous remarquerons dans le troisième chapitre que cette exception leur sera parfois profitable malgré tout¹⁸⁵. Cette exception prévoit qu' « *en l'absence de dispositions contractuelles particulières, ne sont pas soumis à l'autorisation du titulaire les actes visés à l'article 5, a) et b), lorsque ces actes sont nécessaires pour permettre à la personne ayant le droit d'utiliser le programme d'ordinateur, de l'utiliser d'une manière conforme à sa destination, en ce compris la correction d'erreurs* ». Cette exception, dite d'utilisation, permet en d'autres termes, aux personnes bénéficiant de l'autorisation d'utiliser un programme, c'est-à-dire aux licenciés et aux personnes légitimement en possession d'un exemplaire du programme¹⁸⁶, d'accomplir des actes de reproduction techniquement¹⁸⁷ nécessaires à l'utilisation dudit programme, et d'en corriger les éventuelles erreurs. Cela couvre, entre autres, les copies temporaires nécessaires au chargement et à l'exécution des programmes d'ordinateurs de tous types.

Selon l'article 6, § 1^{er} cette exception est supplétive : il s'agirait d'analyser les termes de la licence applicable afin d'apprécier si l'auteur a interdit les actes spécifiques visés. La doctrine insiste cependant sur le fait qu'il faut également tenir compte du considérant n°17 de la Directive 91/250/CEE confirmant que « *les opérations de chargement et de déroulement nécessaires à l'utilisation d'une copie d'un programme légalement acquis, ainsi que la correction de ses erreurs, ne peuvent pas être interdites par contrat* »¹⁸⁸. Les activités de chargement (*loading*), déroulement (*running*) et correction (*debugging*) feraient dès lors l'objet d'une exception impérative.

129. Le concept posant évidemment problème dans l'énoncé de l'article 6, §1 est celui de la destination du programme. La question est de savoir comment déterminer cette destination, et dans quelle mesure on peut la limiter contractuellement¹⁸⁹.

¹⁸⁵ Entre autres dans le cadre des débats sur la viralité des licences et le « linkage », *infra*, n° 135 et s.

¹⁸⁶ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 226.

¹⁸⁷ Voir en ce sens le considérant 17 de la Directive.

¹⁸⁸ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 226 ; F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 224 et s.

¹⁸⁹ T. TJONG TJIN TAI, *op. cit.*, p. 60 et s.

Selon Alain Strowel et Estelle Derclaye, « *La destination résulte de la nature du programme en cause, mais elle peut être déterminée par des dispositions contractuelles particulières (nombre d'utilisateurs, nombre de terminaux, localisation, type d'équipement, etc.), ainsi que par des précisions quant à la fonction du programme (traitement de texte, contrôle de machine-outils, etc.) ou encore par des documents publicitaires* »¹⁹⁰. Ainsi, la destination résulte de la nature du programme, mais peut être déterminée autrement, entre autres, par des clauses contractuelles. Qu'en est-il des dispositions contractuelles qui limiteraient la destination « naturelle » du programme ? La question reste ouverte.

Selon Fernand De Visscher et Benoît Michaux, la destination du programme peut être définie dans le contrat¹⁹¹. Cependant, comme le soulignent André et Henri-Jacques Lucas, « *définir la portée d'une exception, donc le champ du droit exclusif opposable à tous, par référence à des stipulations contractuelles [...] ne s'inscrit pas dans la logique du droit de distribution sujet à épuisement que consacre la directive de 1991, et constitue une innovation discutable* »¹⁹².

La doctrine hollandaise se divise sur le caractère subjectif ou objectif de cette destination¹⁹³. Certains pensent, en effet, que c'est la volonté subjective des parties qui déterminera la destination du programme, même si cette dernière devra s'« objectiver » par les usages et habitudes en cas d'espèces comparables¹⁹⁴. D'autres estiment qu'il s'agit d'interpréter l'exception de manière large et objective, la destination d'un programme étant ses possibilités et capacités intrinsèques, telles que connues par un acheteur expert ou telles que décrites dans une publicité¹⁹⁵. En d'autres mots, selon cette deuxième vision, seule la nature du programme détermine sa « destination ».

Il nous semble qu'il faut trancher en faveur de l'objectivité du critère de destination¹⁹⁶. Partageant la vision d'André et Henri-Jacques Lucas, il nous semble que permettre au titulaire de droits exclusifs de déterminer lui-même l'étendue de l'exception qui altère ses droits ne rime à rien. L'on pourrait rétorquer qu'étant donné que l'exception est supplétive, l'auteur, ayant la faculté d'en démunir purement et simplement son licencié, pourrait *a fortiori* restreindre

¹⁹⁰ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 226.

¹⁹¹ F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 225.

¹⁹² A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 270.

¹⁹³ T. TJONG TJIN TAI, *op. cit.*, p. 61.

¹⁹⁴ P.C. VAN SCHELVEN et H. STRUIK, *Softwarerecht*, Deventer, Kluwer 1995, p. 83 et s.

¹⁹⁵ D.F.W. VERKADE in H. FRANKEN, H.W.K. KASPEREN. et A.H. DE WILDE (red.), *Recht en Computer*, 3^{ème} éd., Deventer, Kluwer, 1997, p. 198 et s.

¹⁹⁶ Dans ce sens, voir S. DUSOLIER, *op. cit.*, 2004, p. 533 : « *Seule une approche objective nous paraît justifiée. C'est dans la normalité de l'utilisation, appréciée au regard de critères objectifs, qu'il faut chercher la conformité à la destination du programme* ».

cette exception comme bon lui semble. Cependant, comme nous l'avons vu, en ce qu'elle concerne les activités de *loading* et *running*, l'exception doit être considérée comme étant impérative. Or ces deux activités nous semblent précisément nous aiguiller sur ce qu'est la nature du programme. Cette nature – ou destination – du programme, c'est, à notre sens, d'être chargé et lancé afin de permettre à l'utilisateur de jouir de ses fonctionnalités et capacités intrinsèques.

Prenons par exemple un système d'exploitation : ses capacités intrinsèques sont notamment la gestion des ressources et du hardware d'une machine, l'échange d'informations entre les unités internes et au final, la préparation de la machine pour que des applications puissent y être lancées. L'application de l'exception interdirait toute restriction contractuelle à ces activités (par exemple, des restrictions portant sur l'installation de certaines applications, la gestion de certains éléments de hardware ou du type de hardware, l'usage par certaines applications des fonctionnalités offertes par le système d'exploitation,...) Par contre, des clauses visant par exemple le nombre de postes sur lesquels l'OS peut être installé ou le nombre d'utilisateurs simultanés ne portent pas atteinte à la nature même du programme, mais relèvent exclusivement du droit de reproduction de l'auteur et, de ce fait, sont acceptables selon nous.

Section 4 : Exercice des droits patrimoniaux : la licence

§ 1. Licences propriétaires et licences libres

130. La licence est le contrat¹⁹⁷ par lequel l'auteur, tout en conservant l'entière propriété intellectuelle sur son œuvre, concède à un licencié l'autorisation de poser sur cette dernière certains actes couverts par ses prérogatives exclusives aux conditions qu'il détermine.

Les licences propriétaires sont généralement octroyées en échange d'un prix, dont le montant est d'ordinaire proportionnel aux droits concédés. Ceux-ci sont accordés parcimonieusement, afin de garder un maximum de contrôle sur la reproduction de l'œuvre, son évolution et sa distribution. Qu'elles soient licences « utilisateur » – ne permettant généralement que l'usage d'une copie du programme sur un poste et/ou par un utilisateur –, licences « distributeurs » – permettant de reproduire et d'installer le programme sur les ordinateurs

¹⁹⁷ Nous ne nous pencherons pas sur les aspects contractuels des licences libres, qui sont abordés par Fabrice DE PATOUL et Yorick COOL dans leurs contributions au présent ouvrage, mais il nous semble cependant opportun d'analyser dans le cadre de cette section certaines particularités de ces licences libres au regard de leur inclusion dans une systématique tout à fait singulière.

distribués –, ou autres, ces licences ne sont généralement pas accompagnées de codes sources.

Les licences libres ont un objectif tout à fait opposé et se distinguent par leur permissivité. Ces licences permettent à leurs bénéficiaires d'utiliser le programme dans n'importe quel but, de l'étudier et l'adapter, de le copier et d'en redistribuer des copies, modifiées ou non¹⁹⁸. Cette autorisation n'est soumise à aucune condition d'ordre financier et, qui plus est, les licences prévoient souvent explicitement l'accès aux codes sources. Par contre, le bénéficiaire doit respecter certaines obligations plus ou moins étendues selon la licence.

Etant donné l'objectif du mouvement libre et les quatre libertés qu'il promeut¹⁹⁹, les licences libres doivent permettre au licencié de poser l'ensemble des actes qui relèvent des prérogatives exclusives de l'auteur, à savoir la reproduction permanente ou provisoire, la traduction, l'adaptation, l'arrangement et toute autre transformation ainsi que la distribution au public et la communication au public.

§ 2. *Un problème de langue*

131. A cet égard, un problème qui se pose est celui de la langue des licences libres et du système légal qui influence leur rédaction. La plupart des licences libres rencontrées sont rédigées en anglais, reprenant les termes et catégories juridiques du droit anglo-saxon. Si les licences prévoient des dispositions concernant la copie, la modification et la distribution, elles parlent aussi parfois d'actes tels que « display to the public » ou « perform to the public ». Le droit de communication au public n'est dès lors jamais, en tant que tel, mentionné. D'autre part, il faut également noter que l'étendue des catégories juridiques diffère. Alors qu'en France ou en Belgique, le droit de distribution ne concerne que les supports physiques, le droit de distribution américain inclut également la mise à disposition des œuvres sur Internet.

Ce problème pourrait être facilement écarté par une interprétation large et téléologique des licences. Cette solution semble néanmoins difficilement « conciliable », dans la mesure où elle s'appliquerait²⁰⁰, avec la règle

¹⁹⁸ Nous renvoyons aux quatre libertés telles que définies par la F.S.F.

¹⁹⁹ Voir note précédente.

²⁰⁰ Cette règle ne s'applique en effet qu'en cas de doute sur la commune intention des parties. Voir A. STROWEL et E. DERCLAYE, *op. cit.*, p. 106 ; H. VANHEES, « Auteurscontractenrecht : actuele situatie en toekomstperspectieve », *R.W.*, 1993-1994, p. 283. Ce qui est confirmé par la jurisprudence, voir par exemple civ. Bruxelles, 12 novembre 1993, *J.L.M.B.*, 1995, p. 918 ; Bruxelles, 25 février 1999, *I.R.D.I.*, 1999, p. 93.

d'interprétation stricte des dispositions contractuelles relatives au droit d'auteur et à ses modes d'exploitation²⁰¹. En définitive, la position du juge qui devra trancher pareille question s'avèrera délicate²⁰².

§ 3. *Diversité de licences*

132. Chaque licence étant différente dans sa façon d'énoncer et de détailler les droits concédés, une lecture attentive et rigoureuse au cas par cas s'imposera.

Les obligations imposées au licencié sont encore plus diversifiées et il nous est impossible de répertorier chacune d'entre elles. Cependant, la contre-partie la plus communément exigée du licencié est le respect des notices de copyright, de crédit et autres clauses de non-garantie. Cette obligation s'inscrit bien évidemment dans une logique de promotion de la renommée des auteurs et de licences gratuites.

Enfin, l'étendue des obligations du licencié dépendra entre autres du caractère copyleft ou non-copyleft de la licence. Une licence non-copyleft ne comportera généralement pas beaucoup plus de conditions en faveur de son auteur ou de tiers bénéficiaires. A l'inverse, la licence copyleft obligera le licencié à distribuer tout programme reprenant une partie du code du logiciel libre selon des termes et conditions similaires ou sous la même licence.

§ 4. *Compatibilité de licences*

133. Il nous semble opportun d'évoquer une problématique difficilement abordable en termes généraux, et qui devra également s'étudier au cas par cas : la compatibilité entre licences. Ce terme, issu du monde du Libre, désigne la possibilité, soit de créer un logiciel sur la base d'une combinaison de codes distribués sous des licences différentes, soit, lorsque du code libre est utilisé, de passer à une licence autre que celle sous laquelle ce code est distribué.

Par exemple, lorsque la F.S.F. estime que la licence B.S.D. (modifiée) est compatible avec la licence G.P.L.²⁰³, cela signifie que tout code sous B.S.D. peut, soit être inclus dans un programme qui sera redistribué sous G.P.L., soit

²⁰¹ Article 3, §1, al. 4. de la LDA. Voy. *infra*, n° 184 et s.

²⁰² Que penser par exemple, au regard du droit belge, de l'article 0 de la G.P.L. prévoyant : « *Activities other than copying, distribution and modification are not covered by this License; they are outside its scope* ».

²⁰³ La F.S.F. établit sur son site une liste des licences qu'elle estime compatibles ou non avec la G.P.L. Voir <http://www.gnu.org/licenses/license-list.html>.

être ajouté à du code distribué sous G.P.L., pour créer un programme qui sera distribué sous G.P.L.

Nous insistons sur le fait que, malgré l'acception traditionnelle du terme, la compatibilité n'est pas réciproque. Alors que la B.S.D. est compatible G.P.L., la G.P.L. ne sera compatible avec aucune autre licence (sauf peut-être, et espérons-le, avec les futures versions de cette dernière), puisque par définition, tout logiciel incluant du code G.P.L. doit être redistribué sous G.P.L. Bien entendu, se cache sous le terme « compatibilité de licences » un agencement complexe d'obligations successives, issues de contrats différents, dont chacune doit être respectée. Pour déterminer si une licence est compatible avec une autre, il faudra donc analyser si aucune obligation de la première n'empêche le respect d'une quelconque obligation de la seconde.

§ 5. *Le « multiple licensing »*

134. Pour terminer, il nous faut souligner une pratique courante en matière de logiciel libre, à savoir le « multiple licensing »²⁰⁴. Les auteurs d'œuvres primaires, tout en étant tenus de respecter les termes des licences libres déjà concédées, par nature non-exclusives, conservent tous leurs droits d'auteur sur leurs œuvres. Ils sont dès lors à tout instant libres de distribuer leur logiciel, concomitamment ou non, sous toute autre licence, y compris sous licence commerciale ou propriétaire.

L'article 13 de la M.P.L. version 1.1 insiste expressément sur cette possibilité laissée à l'auteur originaire de distribuer tout ou partie du programme sous une autre licence. Selon cet article, l'utilisateur a alors l'opportunité de choisir la licence qui lui convient.

La firme MySQL s'est rendue célèbre par cette pratique de double licence. Le principe est de distribuer sa base de données sous G.P.L., afin de contribuer au mouvement libre. Cependant, consciente que certains développeurs continuent à vouloir développer des logiciels propriétaires et ne sont, par conséquent, pas prêts à respecter les termes de la G.P.L., MySQL leur laisse la possibilité d'intégrer tout ou partie de cette base de données dans leurs logiciels destinés à une distribution propriétaire en leur concédant à cet effet une licence commerciale « traditionnelle »²⁰⁵.

²⁰⁴ Voir L. ROSEN, *op. cit.*, 2004, p 262 et s.

²⁰⁵ Voir http://dev.mysql.com/doc/mysql/fr/Business_Services_Commercial_licenses.html.

Enfin, le « multiple licensing » est souvent utilisé afin de prévenir les freins à l'utilisation dus aux problèmes de compatibilité de licence. En effet, nous avons vu que le principe de compatibilité de licence créait des situations où plusieurs codes ne peuvent être assemblés dans un même programme car les termes des licences sous lesquelles ils sont respectivement distribués ne le permettent pas. Par exemple, il est impossible d'ajouter du code sous G.P.L. à du code sous O.S.L. au regard de leur clauses copyleft respectives (selon la G.P.L., le programme résultant devrait être distribué sous G.P.L., alors que d'après les termes de l'O.S.L., le programme résultant devrait être distribué sous O.S.L.). Pour éviter ce genre de situations, l'auteur originaire du programme distribué sous O.S.L. peut dès lors décider de distribuer le même programme sous G.P.L., afin que tout ou partie de celui-ci puisse également être mélangé à du code O.S.L. (et/ou inversement).

CHAPITRE 3 : VIRALITÉ ET CAS D'ESPÈCE PARTICULIERS

135. Dans les deux premières sections, nous avons eu l'occasion d'analyser la naissance des droits d'auteur sur un logiciel, leur contenu, ainsi que la façon dont on pouvait les exploiter, entre autres par le biais de licences.

Nous avons remarqué que les licences libres étaient très atypiques et conditionnaient l'autorisation octroyée par l'auteur au respect de certaines obligations, plus ou moins étendues selon les types de licence. Parmi ceux-ci se distinguent les licences copyleft, dont l'objectif avoué est de se propager de programme en programme afin d'assurer l'avancée du « libre » et l'élargissement de son domaine.

La présente étude serait dès lors incomplète si nous ne nous attardions pas sur certaines questions, aussi débattues que sensibles, relatives aux licences copyleft et à leur « viralité ».

En général, la « viralité » des logiciels libres découle du fait que certaines licences copyleft tentent d'étendre au maximum leur emprise, y compris à tout programme qui serait lié (« lié ») au logiciel libre qu'elle accompagne²⁰⁶. Nous aborderons d'abord le cas de différentes œuvres informatiques caractérisées par la relation d'interaction ou de dépendance qui les lient à

²⁰⁶ K.J. KOELMAN, « Terug naar de bron : open source en copyleft », *Informatierecht/AMI* 2000-8, p. 149-155 : « Dit wordt plastisch ombeschreven als 'besmetting' ('the G.P.L. is viral', of: als een virus). De onder de G.P.L. vallende programma's 'besmetten' de ermee verbonden software ».

d'autres programmes informatiques, à savoir les librairies²⁰⁷, les *drivers* et les *plugins*.

Ensuite, nous aborderons brièvement d'autres cas de figure que nous assimilons à la tendance virale des licences copyleft, tels que la reproduction systématique de code et les systèmes embarqués.

Section 1 : Le « linkage » statique ou dynamique

§ 1. Notions

136. A ce stade, certaines clarifications d'ordre technique s'imposent.

Une librairie informatique est une sorte de boîte à outils informatique. Ce n'est pas un programme en tant que tel, mais plutôt un ensemble de fonctionnalités dont d'autres programmes peuvent se servir ou auxquelles ils peuvent faire appel à leur guise²⁰⁸. On peut distinguer deux sortes de librairies, en fonction du fait qu'elles sont compilées ou non²⁰⁹. La principale différence entre ces deux types de librairies est que les premières permettent d'intégrer la fonctionnalité visée dans le code source même du logiciel, alors que les secondes constituent des fichiers séparés, ce qui implique dès lors qu'un lien soit établi entre le logiciel et la librairie (le « linkage »)²¹⁰. Les librairies compilées sont également appelées librairies programmes (*program library*).

Parmi les librairies compilées, une distinction fondamentale doit être opérée entre les librairies statiques (*static libraries*) et dynamiques (*dynamic libraries*).

²⁰⁷ De l'anglais *library*, signifiant « bibliothèque ». Il appert que le monde informatique francophone préféra franciser ce terme technique plutôt qu'en respecter sa traduction littérale.

²⁰⁸ Prenons par exemple le code qui dirige la luminosité des pixels d'un écran. Presque tout programme interagit avec son utilisateur par le biais d'interfaces qui apparaissent à l'écran. Ce code ordonnant à la machine d'illuminer tels et tels pixels est très complexe et fastidieux et s'apparente fortement à du code machine (« code objet »). Afin d'éviter aux informaticiens de s'arracher les cheveux et d'encoder ce code dans leur programme, cette fonctionnalité est incluse dans une librairie et peut ainsi être appelée à tout moment.

²⁰⁹ Généralement des packages concernant l'un ou l'autre langage de programmation (pour le « C », citons Visual C, par exemple) sont disponibles. Ceux-ci contiennent un éditeur, une librairie non-compiler et un compilateur. Grâce à cet outil, les programmeurs peuvent éditer du code, y insérer le code de fonctionnalités « prêt-à-porter » trouvé dans la librairie et compiler le tout.

²¹⁰ Le linkage est la phase finale de compilation. Après avoir compilé le code source en code objet, le « linkage » consiste à établir les liens nécessaires entre le logiciel et les librairies dont il se sert, afin, au final, de créer un fichier exécutable.

Au moment du « linkage », c'est-à-dire du passage du code objet au code exécutable, le code de la librairie statique correspondant aux fonctions appelées est intégralement repris dans le logiciel en devenir. Ce dernier consiste dès lors en un programme exécutable autonome. A l'inverse, le linkage n'a pas pour résultat l'intégration des fonctionnalités d'une librairie dynamique dans le code exécutable du logiciel. Dans ce cas, seule une référence est inscrite dans le code exécutable, qui, durant son exécution, appellera la fonction désignée dans la librairie, ce qui aura pour conséquence le chargement de cette dernière dans la mémoire vive de l'ordinateur. Le fichier exécutable ainsi obtenu n'est donc pas autonome et nécessite la présence du fichier librairie à ses côtés²¹¹.

137. Les *drivers* ou *device drivers* sont des petits fichiers nécessaires au bon fonctionnement de tout élément de hardware connecté à l'ordinateur. Ils permettent l'interaction entre les logiciels (entre autres, le système d'exploitation) et le hardware. Les *drivers* sont comparables aux librairies dans le sens où certains seront statiques²¹² (c'est-à-dire chargés systématiquement au démarrage de l'ordinateur) alors que d'autres seront dynamiques (c'est-à-dire chargés lorsqu'un programme le requerra)²¹³.

138. Les *plugins* sont des logiciels qui doivent interagir avec d'autres logiciels afin de remplir une fonction spécifique, complétant généralement les fonctionnalités du logiciel avec lequel ils sont liés²¹⁴. Ces logiciels sont encore un exemple de programmes impliquant un linkage dynamique.

§ 2. Raisonement juridique général

139. La question juridique particulière posée par les librairies, *drivers* et *plugins* est de savoir dans quelle mesure les lier avec un logiciel entraîne un effet de contamination « copyleft », au cas où l'un de ces éléments ou le logiciel

²¹¹ L'exemple le plus connu de librairie dynamique est le fichier DirectX de Windows. Bon nombre d'applications Windows requièrent ce programme et c'est la raison pour laquelle la version de DirectX est mentionnée dans les configurations minimales nécessaires à leur exécution. Sans la bonne version de DirectX le programme ne pourra pas fonctionner car il ne trouvera pas les fonctionnalités nécessaires dans la librairie.

²¹² Les *drivers* - ou modules en langage « Linux » - peuvent normalement tous être chargés dynamiquement ou statiquement, selon le choix de la personne qui installe le système d'exploitation. Cependant, les *drivers* essentiels, tel que le *driver* d'une carte mère par exemple, seront généralement installés en *static loading*.

²¹³ Le *driver* d'une imprimante constitue un bon exemple de *driver* installé dynamiquement.

²¹⁴ Les *plugins* sont souvent rencontrés en matière de multimédia. Ainsi un navigateur web aura besoin d'un *plugin* pour pouvoir exécuter une page web en Flash, ou Shockwave. De même, il est possible d'installer des *plugins* complétant le média-player afin de pouvoir visionner des fichiers audio-visuels d'un format spécial, tel que le DivX par exemple.

en question est distribué sous licence libre de ce type. Certaines licences copyleft²¹⁵ précisent en effet que tout logiciel « basé sur le programme » auquel elles s'appliquent ou « dérivé de ce dernier » doit également être distribué sous cette licence. La question à laquelle nous tenterons d'apporter des éléments de réponse est dès lors de savoir dans quelle mesure ce genre de clause est également applicable au linkage. Plus fondamentalement, nous nous demanderons si, en droit belge, cette prérogative de conditionner le linkage avec un logiciel fait effectivement partie des droits exclusifs de son auteur.

140. La question classique et récurrente concerne par exemple le fait de lier un logiciel avec une librairie distribuée sous licence G.P.L. A la lecture des termes de la licence L.G.P.L., on se rend immédiatement compte que la F.S.F.²¹⁶ estime que tout programme faisant appel aux fonctions d'une librairie est un logiciel dérivé de cette librairie, que cette dernière soit statique ou dynamique et que dès lors ce logiciel ne peut être distribué que sous G.P.L.²¹⁷. Le principal auteur de

²¹⁵ Les licences G.P.L. et L.G.P.L. entre autres.

²¹⁶ Pour rappel, la Free Software Foundation est l'auteur des licences G.P.L. et L.G.P.L., la seconde étant la version affaiblie de la première d'un « point de vue copyleft ». *Supra*, n° 6.

²¹⁷ Le paragraphe 10 du préambule de la L.G.P.L. déclare: « *When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library* ». L'article 5 de la L.G.P.L. précise: « *A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License. However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables. When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law. If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6). Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.* » Il faut noter que la F.S.F. n'est pas la seule à estimer que le linkage dynamique nécessite autorisation de l'auteur du programme lié ; ainsi l'article 6 de la Q.P.L. « régule » le linkage avec le programme auquel elle s'applique. A l'inverse, la licence Apache (ver. 2.0) prévoit explicitement que la notion de « derivative work » telle qu'y définit ne comprend pas les programmes liés.

la licence G.P.L., Eben Moglen, ne manque d'ailleurs pas de confirmer ce point de vue²¹⁸.

141. Au regard du droit américain, il faut cependant d'emblée souligner que la notion d'œuvre dérivée telle que retenue dans la G.P.L.²¹⁹ est différente de celle donnée par la loi américaine (17 United States Code § 101)²²⁰, et que la définition de la G.P.L. ne semble pas inclure le « linkage »²²¹. Par ailleurs, tous ne partagent pas l'avis d'Eben Moglen. En effet, pour Lawrence Rosen, membre actif et conseil de l'O.S.I. (*supra*, n° 9), pour qu'un programme soit une œuvre

²¹⁸ « *The situation is no different than the one where your code depends on static or dynamic linking of a G.P.L.'d library, say GNU readline. Your code, in order to operate, must be combined with the G.P.L.'d code, forming a new combined work, which under G.P.L. section 2(b) must be distributed under the terms of the G.P.L. and only the G.P.L.* ». E. MOGLEN, interviewé par Slashdot, 20 février 2003, voir <http://interviews.slashdot.org/interviews/03/02/20/1544245.shtml?tid=117&tid=123>.

Dans ce sens, Jason B. Washa présente cette vision comme étant une pratique communément acceptée. Cependant l'auteur semble également accepter la pratique selon laquelle des drivers dynamiques sous linux ne doivent pas forcément être distribués sous G.P.L. Voir J.B. WASHA, « Open Source, Free Software, and the General Public Licence », *The Computer & Internet Lawyer*, Mars 2003, n° 3, p. 22 et s.

²¹⁹ L'article 0 de la G.P.L. donne la définition suivante : « The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) »

Il faut cependant ajouter à cette définition le contenu de l'article 2, b de la G.P.L., qui précise : « You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

[...]

b) You must cause any work that you distribute or publish, that in whole or in part contains or is **derived from the Program or any part thereof**, to be licensed as a whole at no charge to all third parties under the terms of this License » (nous soulignons).

Beaucoup voient en cette dernière phrase la source de la viralité de la G.P.L., interprétant le terme « derived from » de manière (à notre sens exagérément) extensive. Voir en ce sens : ROSEN, L., « General Public Licence, Explained », *sitepoint.com*, 23 may 2001, voir <http://www.sitepoint.com/article/public-licence-explained>.

²²⁰ Le § 10 du titre 17 du Code Américain définit l'œuvre dérivée comme étant « *A work based upon one or more pre-existing works such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work or authorship, is a 'derivative work'* ».

Sur la différence entre la définition de la G.P.L. et de la loi américaine, voir J.B. WASHA, *op. cit.*, p. 22. ; Ph. LAURENT, *op. cit.*, p. 30.

²²¹ Il est en effet bien marqué « containing the program or a portion of it », et non pas, par exemple, « containing and/or linking with the program or a portion of it ».

dérivée d'un logiciel libre (dans le cas d'espèce, d'une librairie) et tombe dès lors sous l'emprise du copyleft, il faut qu'il y ait modification du code du programme afin que le lien soit correctement effectué²²². A l'inverse, toujours selon Lawrence Rosen, le linkage dynamique n'est qu'une relation transitoire entre deux programmes, pour laquelle ils sont prédestinés. Dans ce cas d'espèce, le programme faisant appel à la librairie ne doit pas être modifié afin que le linkage s'effectue²²³. Si nous ne sommes pas convaincus par l'aspect technique de cette dernière explication, l'aspect juridique du raisonnement nous paraît fondé.

142. Les notions d'œuvres dérivées en droit américain, français et belge sont comparables. Cependant, la définition légale française, reprise par la doctrine belge, est plus précise dans la mesure où elle prévoit explicitement l'« *incorporation d'une oeuvre préexistante sans la collaboration de l'auteur de cette dernière* », alors que la définition américaine repose sur les termes, à notre sens plus vagues, d'œuvre basée sur une œuvre préexistante (« *work based upon one or more pre-existing works* »). Dès lors, si le doute est concevable outre-Atlantique, il ne l'est pas en droit belge et français : pour qu'il y ait œuvre dérivée, il faut qu'il y ait reprise totale ou partielle du code du programme original.

143. Dès lors, la distinction entre le linkage dynamique et statique est fondamentale sur le plan juridique et ce, aussi bien en matière de librairies que de *drivers* ou *plugins*. Comme nous l'avons vu, le linkage statique entraîne reprise du code de la librairie dans la version finale exécutable du logiciel. Le code binaire du logiciel incorporant une partie du code de la librairie, ce logiciel doit être considéré comme étant une œuvre dérivée de cette librairie. Si cette dernière est distribuée sous G.P.L., le logiciel résultant devra également être distribué sous G.P.L. Cette solution vaut également pour les *drivers* en *static loading*²²⁴.

En matière de linkage dynamique, les choses sont loin d'être si simples. Ce genre de linkage a pour particularité que les fonctionnalités de la librairie utilisées par le logiciel ne sont pas incorporées à la version exécutable de ce

²²² L. ROSEN, *op. cit.* : « *Software experts distinguish among various forms of program linking to create a combined program. Static linking requires a modification to the code of one program to allow it to link to another program. Such a modification, since it requires changes to the source code of the linking program, arguably creates a derivative work. If the linking program is licensed under the G.P.L., then the derivative work also becomes subject to the G.P.L.* ».

²²³ L. ROSEN, *op. cit.* : « *Dynamic linking, on the other hand, is a transitory relationship between two programs for which they are each predesigned. The linking program need not be modified to implement the linkage* ».

²²⁴ J.B. WASHA, *op. cit.*, p. 23.

dernier. Par contre, l'exécution du programme nécessite la présence du code objet de la librairie à ses « côtés », afin qu'il puisse y appeler lesdites fonctions. Dès lors, la fusion entre les deux codes n'est effectuée que lors du chargement simultané du logiciel et de la librairie en mémoire vive. Comme le souligne Lawrence Rosen, il s'agit d'une relation transitoire entre les deux programmes. Le code des fonctionnalités des librairies n'est donc pas repris dans le logiciel, ni dans son code source, ni dans le code objet, ni dans le code exécutable. Dès lors, le logiciel n'est pas, à première vue, à considérer comme étant une œuvre dérivée de la librairie. Le programmeur, en développant ce logiciel, n'a posé aucun acte relevant des droits exclusifs de l'auteur de la librairie et n'a besoin d'aucune autorisation de l'auteur de cette dernière²²⁵. Il en découle que si la librairie est distribuée sous G.P.L., le logiciel qui y est dynamiquement lié pourra être distribué selon les termes et conditions choisis en toute liberté par son auteur. Ce raisonnement appelle cependant plusieurs réserves ou contre-arguments de poids.

§ 3. Réserves

144. Une première réserve est fondée sur la nature même du linkage dynamique. S'il est vrai que le code de la librairie n'est pas repris dans le code du logiciel y faisant appel, cet appel ne peut être fait que par référence, ce qui entraîne reprise dans le code du logiciel de ce que nous appellerons des « identifiants » propres à la librairie, permettant à l'ordinateur de retrouver les fonctions utilisées au sein de la librairie et de les appliquer correctement en fonction des besoins du logiciel²²⁶. L'on pourrait dès lors prétendre que ces identifiants, à savoir le nom de la librairie et de ses fonctions, sont protégés par le droit d'auteur et que leur reprise dans le code du logiciel nécessite

²²⁵ L'on pourrait sans doute imaginer insérer dans les licences libres une clause permettant l'utilisation du programme à condition que le licencié ne crée aucun logiciel qui y soit dynamiquement lié sans être distribué sous les mêmes termes et conditions que la licence concédée. Le licencié qui manquerait à cette obligation engagerait dès lors sa responsabilité contractuelle. Cependant, l'acte posé n'est pas une violation du droit d'auteur du titulaire mais une simple contravention à la convention-loi. L'on peut également mettre en question la validité et l'applicabilité de cette clause. Enfin, notons qu'il n'est pas impossible de créer un programme lié à une librairie sans reproduire tout ou partie de cette dernière (il suffit en effet simplement de connaître ses identifiants).

²²⁶ En pratique, cela implique :

- une reprise du nom de fichier de l'API (Application Program Interface – sorte de menu des fonctions de la librairie et mode d'emploi permettant l'interfaçage entre celle-ci et le logiciel) dans le « header » (sorte d'entête explicative) ainsi qu'un appel des fonctions par le biais de symboles (les noms de ces fonctions) dans le code source du logiciel ;
- la mention du nom de la librairie et sa localisation lors du linkage ;
- que l'exécutable quant à lui, contienne le nom de la librairie, sa localisation, ainsi que les références telles qu'effectuées ci-dessus.

l'autorisation de leur auteur. L'on pourrait également considérer ces identifiants comme faisant partie de l'interface permettant la communication entre le logiciel et la librairie. Ils bénéficieraient dès lors, selon Alain Strowel et Estelle Derclaye, de la protection par le droit d'auteur dans la mesure où ils constituent une forme originale d'expression²²⁷. Cependant, en droit d'auteur traditionnel, la doctrine admet qu'il est difficile de reconnaître la protection par le droit d'auteur à quelques syllabes ou à quelques mots²²⁸ et la jurisprudence s'est toujours montrée hésitante à ce sujet²²⁹. De plus, la Cour de cassation française estime qu'un titre, s'il bénéficie de la protection par le droit d'auteur, n'est cependant pas protégé contre tout usage, et entre autres, contre son utilisation dans un index documentaire²³⁰. L'on en conclut dès lors que la protection des identifiants d'une librairie est loin d'être garantie, spécialement au vu de l'objectif de leur reprise dans le code du logiciel.

145. Même si tout ou partie du code de la librairie n'est pas reproduit dans le code du logiciel (sauf identifiants), l'on pourrait aussi argumenter qu'une réelle « fusion » entre le logiciel et la librairie s'effectue dans la mémoire vive de l'ordinateur de l'utilisateur qui exécute le logiciel. L'on pourrait dès lors prétendre que l'œuvre dérivée se constitue dans la mémoire temporaire des ordinateurs des utilisateurs. Cet argument passe cependant à côté de son objectif. Tout d'abord, une réplique consisterait à dire que cet acte de reproduction n'est pas effectué par l'auteur du logiciel, mais par son utilisateur. Dès lors, dans pareille situation de reproduction, le titulaire des droits sur la librairie devrait s'adresser non pas aux auteurs de logiciels y faisant appel, mais aux utilisateurs

²²⁷ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 199. Il faut cependant noter que les droits d'auteur sur les interfaces sont susceptibles de faire l'objet de licences obligatoires. Nous pensons entre autres à la décision de la Commission européenne dans le cadre de l'affaire « Microsoft » (voir <http://europa.eu.int/comm/competition/antitrust/cases/decisions/37792/en.pdf>).

²²⁸ Voir entre autres A. BERENBOOM, *op. cit.*, p. 73 et A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 101.

²²⁹ Par exemple, les cours et tribunaux ont refusé la protection aux titres suivants: « Ali Baba et les 40 voleurs » (Com. Bruxelles, 31 décembre 1953, *R.G.A.R.*, 1954, 5442), « Le Parrain » (TGI Paris, 20 mars 1975, *Ann.* 1976, 55), « Les Maîtres du Temps » (TGI Nanterre, 28 juin 1995, *R.T.P.I.*, n° 62, 52). Par contre la protection fut reconnue à « Le chardon » (CA Paris, 1ère ch., 25 septembre 1989, *R.I.D.A.* 2/1990, p. 207) ou à « L'affreux Jojo » (TGI Paris, 15 juin 1972, *R.I.D.A.* 1975, 457). Notons également que le Tribunal de première instance de Bruxelles a jugé, à juste titre, que les mots-clés placés en introduction de décisions publiées, constituant des renvois obligatoires et non-originiaux vers les matières juridiques traitées, n'étaient pas protégés par le droit d'auteur. (Civ. Bruxelles, 28 Juillet 2000, *R.W.*, 2002-2003 p. 829 et s., note de H. VANHEES, *A.J.T.*, 2000-2001, p. 10 et s.). Les identifiants de librairies sont d'une certaine mesure comparables à ces mots-clés, dans la mesure où leur utilisation est obligatoire afin d'appeler les fonctions y étant associées.

²³⁰ Cass. Ass. Plén., 30 octobre 1987, *R.I.D.A.*, 1/1988, p. 78, concl. CABANNES, citée par A. LUCAS et H.-J. LUCAS, *op. cit.*, p. 102.

de ces derniers. Ensuite, à supposer que les licences restreignent l'usage des librairies à un linkage purement « libre », encore faut-il tenir compte de l'exception d'utilisation prévue à l'article 6, § 1, de la LPO. Grâce à la licence libre, tous les utilisateurs de la librairie sont acquéreurs légitimes d'un exemplaire de la librairie²³¹ et bénéficient dès lors du droit de reproduire celle-ci sur le disque dur de leur ordinateur et de l'utiliser. En vertu de l'article 6, § 1, de la LPO, et en l'absence de dispositions contractuelles particulières, les utilisateurs n'ont pas besoin de l'autorisation des titulaires de droits d'auteur sur la librairie pour poser des actes visés à l'article 5, a) et b) de la LPO lorsque ces actes sont nécessaires pour leur permettre de l'utiliser d'une manière *conforme à sa destination*. La nature même d'une librairie dynamique, à savoir sa « destination » au sens « objectif » du terme, est d'être chargée en mémoire vive lorsqu'un logiciel fait appel à ses fonctions. La reproduction de tout ou partie de la librairie effectuée en mémoire vive de l'ordinateur semble dès lors couverte par l'exception, dans la mesure où l'on penche pour une application objective du critère de « destination » du programme. En tout état de fait, si l'on penche pour une vision subjectiviste du critère de la « destination », s'agissant de la G.P.L., comme de toutes les autres licences analysées, il nous semble que les termes utilisés sont trop larges et ne soumettent pas explicitement la copie provisoire nécessaire à l'utilisation de librairies aux termes de la licence²³².

146. L'on pourrait également avancer que le linkage dynamique est plus qu'un cas classique de reproduction provisoire en mémoire vive, mais qu'il s'agirait davantage, à défaut d'une incorporation réelle de tout ou partie du code de la librairie dans celui du logiciel, d'une incorporation conceptuelle²³³ : même si les lignes de code de la librairie ne sont pas matériellement reproduites dans celles du logiciel, le logiciel dans son ensemble exécutable est intellectuellement « conçu » en incorporant des éléments de la librairie. Même si elle n'est pas reconnue en tant que telle par la loi, cette notion d'incorporation intellectuelle a déjà germé dans les esprits de certains auteurs confrontés à l'Internet et plus particulièrement aux techniques d'*in-lining* et de *framing*²³⁴. Ces techniques consistent à composer le code de pages web en y incorporant des liens faisant apparaître à l'écran des objets qui proviennent en fait d'autres sites web. Ainsi,

²³¹ Sur les conditions d'application de l'article 6, §1, de la LPO, voir par exemple A. STROWEL et E. DERCLAYE, *op. cit.*, p. 226.

²³² En effet, paradoxalement, c'est dans la L.G.P.L. que la F.S.F. déclare que le copyleft de la G.P.L. est de rigueur même en cas de linkage dynamique.

²³³ Ph. LAURENT, *op. cit.*, p. 32.

²³⁴ S. DUSOLLIER, « Les outils de références : les cartes aux trésors de l'Internet », in E. MONTERO (éd.), *Droit des technologies de l'information – Regards Prospectifs*, Cahiers du CRID, n°16, Bruylant, 1999, p. 33-53. On pourra consulter également l'article de A. STROWEL et N. IDE, « Responsabilité des intermédiaires sur Internet : Actualité et questions des hyperliens », accessible à l'adresse http://www.droit-technologie.org/dossiers/responsabilite_intermediaires_questions_des_hyperliens.pdf.

le navigateur du visiteur de la page compose cette dernière en y incorporant les objets « empruntés », et affiche à l'écran le contenu de la page, agrémenté par exemple, de photos d'autres sites, le visiteur n'ayant par ailleurs pas même conscience de cet emprunt. Si ces techniques sont semblables au linkage dynamique, dans la mesure où il n'y a pas reprise de tout ou partie de l'œuvre dans le code, mais qu'une fusion entre les deux œuvres s'opère au niveau de la mémoire vive de l'ordinateur, la comparaison s'arrête cependant à notre sens aux particularités fonctionnelles que présentent les librairies, ainsi que les éléments chargeables dynamiquement : ils peuvent être interchangeables. En effet, il arrive régulièrement que certaines fonctions se présentent dans plusieurs librairies différentes. Certains programmeurs de logiciels tiennent compte de ce fait, et prévoient dans leur code que si une des librairies n'est pas disponible, le logiciel fera appel à l'autre. Dans ce contexte, l'on se rend compte qu'en définitive, si l'on admettait l'application de la théorie de l'incorporation conceptuelle, l'on aurait du mal à définir de quelle librairie le logiciel est finalement une œuvre conceptuellement dérivée : de l'une, de l'autre, des deux simultanément ? Notre réponse serait de l'une ou de l'autre suivant la librairie qui fut finalement utilisée, mais cette réponse s'accorde mal avec le concept même de reproduction intellectuelle. Plus fondamentalement, il nous semble que, s'agissant du linkage à différentes librairies substituables, c'est l'idée (la fonction envisagée) qui fait l'objet de l'incorporation intellectuelle, peu importe la forme dans laquelle elle fut exprimée (et donc peu importe le fait que la fonction soit appelée dans une librairie ou l'autre). L'incorporation intellectuelle ne porterait dès lors pas, dans le cas d'espèce, sur un objet protégeable en tant que tel par le droit d'auteur.

147. Au vu de ce qui précède, il nous faut dès lors constater qu'aucune base juridique solide et incontestable, puisant sa source dans le droit d'auteur, ne soutient l'affirmation selon laquelle une licence copyleft peut imposer au programmeur qui crée un logiciel faisant appel aux fonctions d'une librairie ou de tout autre élément dynamique distribué sous cette licence, de distribuer son programme sous cette même licence. Cette affirmation ira en effet généralement au-delà de ce que les droits exclusifs de l'auteur lui garantissent comme contrôle sur son œuvre, allant de ce fait également à l'encontre des droits exclusifs propres à l'auteur de la prétendue œuvre dérivée.

Notons enfin qu'une doctrine américaine se base sur la théorie de l'abus de droit pour refuser aux auteurs d'éléments dynamiques cette possibilité de contrôler le linkage²³⁵.

²³⁵ « *Even if the open source license is binding, the copyleft provision may still not be enforceable as to independent proprietary code, in light of the intellectual property misuse doctrine. The doctrine is asserted as an affirmative defense to an intellectual property infringement claim. Much like an unclean hands defense, the misuse doctrine precludes*

Peut être la protection contre le linkage abusif se trouvera-t-elle ailleurs ? Nous pensons entre autres au droit de la concurrence et à ses théories sur le parasitisme.

Section 2 : Autres cas d'espèce

148. Nous avons pu constater que l'aspect viral traditionnellement reconnu à certaines licences copyleft est issu des techniques de linkage et de l'insécurité juridique qui en découle. En définitive, il s'avère que cet effet présumé est exagéré et, du point de vue du droit d'auteur, difficilement fondé.

Par contre il nous semble que l'on pourrait étendre la notion de « viralité » à d'autres situations techniques. Nous aborderons dans le présent point deux cas précis, la reproduction systématique de code et les systèmes embarqués.

§ 1. *Reproduction systématique de code*

149. Jusqu'à présent, nous sommes partis du principe que le code était généré par les programmeurs, soit sur la base de « copier-coller » d'autres lignes de codes, soit sur la base de leur propre programmation originale. Le monde informatique est cependant beaucoup plus complexe, et il existe bon nombre de programmes qui génèrent à leur tour du code informatique. Les auteurs s'accordent à dire que pour qu'une œuvre soit protégée par le droit d'auteur, il faut qu'il y ait intervention de la créativité humaine : un code généré exclusivement par une machine n'est donc pas protégé par le droit d'auteur. Par contre, lorsqu'un programme d'ordinateur générant du code y inclut d'office des lignes de son propre code, et donc issues d'un travail intellectuel humain, des problèmes risquent de se poser.

150. Prenons le cas du programme Bison²³⁶. Sans entrer dans les détails, ce programme est un générateur de code source, un compilateur permettant de traduire un programme d'un langage à un autre, et qui fut pendant un temps distribué sous G.P.L. Ce programme avait pour particularité d'intégrer dans chaque code qu'il générerait une partie non négligeable de son propre code. Dès

enforcement of intellectual property rights that have been extended beyond the scope of those rights ». Ch. H. NADAN, « Open Source Licensing : virus or virtue? », *Texas Intellectual Property Law Journal*, spring 2002, p. 367.

²³⁶ Sur le programme Bison, voir entre autres Ch. DONNELLY et R. STALLMAN, « The YACC-compatible parser generator », http://www.lns.cornell.edu/public/COMP/info/bison/bison_toc.html#TOC2.

lors, selon les termes de la licence G.P.L. applicable à Bison, tout programme créé par Bison devait être soumis à cette dernière.

Craignant que bon nombre de programmeurs ne boycottent Bison à cause de cette contamination²³⁷, ses auteurs décidèrent de modifier les termes de leur licence, afin de permettre que le résultat de leur création puisse rester propriétaire. Ainsi, une personne désireuse de créer un programme avec Bison peut depuis lors décider de distribuer ce programme aux conditions qu'elle détermine.

151. Par cet exemple, on remarque que le linkage n'est pas la seule source de « contamination » technique, et que l'usage de certains outils de programmation distribués sous licence copyleft peut l'être également. Cette nouvelle forme de viralité est cependant beaucoup plus pernicieuse, dans la mesure où le programmeur n'étant pas au courant de cette insertion systématique de code au sein du code objet et qui distribuerait son œuvre sous licence propriétaire violerait les termes de la licence copyleft du compilateur sans même en être conscient. De plus, alors que nous avons vu ci-dessus que la contamination par linkage était loin d'être légalement fondée, la reproduction systématique de code a pour effet que tout programme compilé est d'office une œuvre dérivée du compilateur au sens de la loi belge. L'obligation de distribuer le programme ainsi compilé sous la même licence libre est dès lors certaine.

§ 2. *Systèmes embarqués*

152. Les systèmes embarqués (« embedded systems ») sont des machines généralement usuelles, autres que des PC ou serveurs, et au sein desquelles des logiciels sont implantés. Les exemples types de systèmes embarqués sont les GSM, agendas électroniques, les lecteurs MP3,... L'on a moins l'habitude de citer les missiles, les poupées parlantes, les appareils photos numériques ou les nouvelles générations d'amplificateurs de guitare...

La notion de système embarqué provient du phénomène de la robotisation. Au fil de l'évolution technologique et de sa maîtrise de l'électronique, l'homme intègre de plus en plus des composantes informatiques dans ses créations, et depuis les premières calculatrices et montres électroniques, on en est maintenant arrivé aux complexes appareils miniatures multifonctions.

²³⁷ Puisqu'il y a effectivement reprise de partie de code source, cette application des termes de la G.P.L. est également tout à fait valide en droit belge: tout programme créé par Bison – logiciel distribué sous G.P.L. – doit également être distribué sous G.P.L.

153. La différence classique entre les systèmes embarqués et les systèmes informatiques de type PC est que l'on ne peut généralement pas y distinguer disque dur et mémoire vive. Dès lors, il n'y a généralement pas de processus de chargement en mémoire vive ni d'allocation de mémoire : tous les programmes sont automatiquement chargés et opérationnels au moment du lancement de la machine. De même il est également difficile d'établir au sein des systèmes embarqués une séparation claire entre le système d'exploitation et les programmes d'application. Cette distinction s'amenuise cependant au fil des évolutions technologiques, les nouvelles générations d'appareils numériques présentant actuellement parfois toutes les spécificités des ordinateurs classiques. Dans les développements suivants, nous qualifierons les systèmes embarqués de « simples », lorsqu'ils présentent les caractéristiques ci-dessus définies et les distinguant des PC classiques.

154. De plus en plus de systèmes embarqués « simples » sont conçus sur base du Linux Kernel²³⁸. Le principe consiste généralement à prendre la base minimale du noyau Linux et d'y greffer l'application nécessaire qui contrôlera le système embarqué. Si l'on considère cette méthode de développement et les spécificités des systèmes embarqués « simples », les questions juridiques se posent rapidement. Le code d'un système embarqué « simple » est-il une œuvre unique dérivée de Linux, et donc soumis entièrement aux termes et conditions de la G.P.L., ou faut-il malgré tout établir la distinction classique entre système d'exploitation et application, ce qui aurait pour conséquence que ce dernier pourrait être distribué par son auteur sous la licence de son choix²³⁹ ? Dans la mesure où la partie informatique des systèmes embarqués simples constitue un tout s'exécutant immédiatement et simultanément (faisant par ailleurs corps avec la machine qu'il contrôle), il nous semble que l'on aurait du mal à distinguer deux œuvres séparées dans ce programme de nature unique et indivisible. Le code du système embarqué en question devrait dès lors être soumis entièrement aux termes de la G.P.L., ce qui entraîne de nouvelles questions : comment communiquer les termes de la licence, le code source et les notices comme l'impose la G.P.L. alors que le programme en question est intégré dans un objet utilitaire n'ayant pas forcément d'écran ou d'interface

²³⁸ Voir D. ROSENBERG, « The secret battlefield : Embedded systems and Linux » <http://www.developer.com/tech/article.php/629301>; *Open source, the Unauthorized white papers*, M&T books, 2000, <http://www.stromian.com/Book/FrontMatter.html#Contents> ; R. STEVENSON, « The future of embedded systems », http://www.cyberguard.com/pdf/sea_dec02_stevenson_56_58.pdf.

²³⁹ Il est en effet unanimement admis dans la pratique, et indiscutable d'un point de vue juridique, qu'en ce qui concerne les ordinateurs classiques, le système d'exploitation et les applications sont des œuvres distinctes et légalement indépendantes. Dès lors, il est tout à fait légal de créer des applications « propriétaires » qui tournent sur des OS libres tels que Linux.

visuelle²⁴⁰ ? De plus, à quoi bon pouvoir modifier pareil code source si ce dernier n'est installé qu'en usine et ne peut plus être modifié une fois intégré à l'appareil ? Par ailleurs, dans les cas où il serait possible de modifier le code de pareils programmes et de les mettre à jour²⁴¹, l'on pourrait comprendre la volonté des constructeurs de ne pas dévoiler le code source afin d'éviter que les machines ne dysfonctionnent et, au final, mettent en péril la vie de leurs utilisateurs, ce qui pourrait éventuellement engager leur responsabilité... (nous pensons par exemple aux systèmes de sécurité électroniques présents sur nos voitures actuelles, ou sur des appareils médicaux...) Il nous semble dès lors que les termes et conditions classiques des licences « copyleft » s'accordent mal avec la technique des systèmes embarqués.

155. Dans les systèmes embarqués plus évolués, la question du linkage peut également refaire surface : dans la mesure où, dans bon nombre de cas de systèmes embarqués, il est difficile de distinguer disque dur et mémoire vive, le linkage entre une application et une librairie doit-il toujours être considéré comme étant statique ?

Cette question devra certainement se résoudre au cas par cas, en fonction de la technique utilisée et de reproduction et/ou intégration éventuelle de code, mais nous pouvons déjà à l'heure actuelle prévoir que la tâche du juge sera ardue.

CONCLUSION

156. Le droit, à l'instar de la technologie informatique, ne cesse d'évoluer et de se complexifier. Alors que le premier se veut de plus en plus « technologiquement neutre », la seconde se diversifie et se renouvelle à une vitesse impressionnante, donnant naissance à de nouveaux procédés, de nouvelles catégories, balayant au passage les précédents cloisonnements logiques. Le travail consiste dès lors à encadrer minutieusement l'inconnu et la nouveauté par les critères et concepts juridiques disponibles.

Le mouvement libre n'a pas uniquement créé une nouvelle façon d'appréhender la systématique des droits exclusifs des auteurs, il en a également exploré les frontières dans des sens encore jusqu'ici non-envisagés. Entre la logique propriétaire et la logique libre, le droit d'auteur se voit une fois de plus tiraillé

²⁴⁰ L'on pourrait peut-être argumenter que les programmes présents au sein des systèmes embarqués ne sont pas « mis à la disposition du public », mais uniquement les résultats qu'ils produisent. En effet, l'on pourrait considérer que ces logiciels embarqués ne sont pas toujours, en tant que tels, accessibles au public.

²⁴¹ Par exemple, grâce à la technologie PROM (Programmable ROM) ou EPROM (Erasable and Programmable ROM).

par des intérêts contradictoires, sa couverture tirée de chaque côté du lit, sans trop savoir à quel point celle-ci est extensible.

Cependant, il nous semble que le droit d'auteur, par ses principes fondamentaux et son équilibre sous-jacent, permettra toujours au juriste de trouver des solutions justes et fondées sur des raisonnements juridiques valides, pour peu qu'il s'arme de patience, qu'il prenne la peine de comprendre et d'assimiler la technique, et qu'il soit sensible aux différences de conception et de logique qui le séparent des informaticiens.

Même si l'exercice s'avère rigoureux, une analyse de la chaîne d'œuvres dérivées que constitue l'évolution du logiciel permet de déterminer les titulaires de droits d'auteur sur les œuvres subséquentes, ainsi que les personnes dont l'autorisation est requise pour la création d'œuvres dérivées futures. Cette identification est évidemment essentielle afin d'apprécier la validité et le crédit d'une licence. A l'inverse, en cas de litige, elle sera également d'utilité pour déterminer les personnes effectivement aptes à réclamer le respect des termes de la licence utilisée ou à intenter une action en justice.

Quant aux droits patrimoniaux et aux exceptions y afférentes, ils sont les mêmes pour tous les programmeurs, qu'ils soient du camp libre ou du camp propriétaire. Afin d'en respecter l'équilibre, le droit d'auteur se doit d'être appliqué de façon neutre et intransigeante, et ce, même si les objectifs et l'esprit des titulaires de droits peuvent paraître opposés. Le système copyleft, s'il se verra légèrement limité (entre autres son aspect viral) par pareille application du droit d'auteur, gagnera selon nous en sécurité juridique, ce qui, au final, permettra aux développeurs dans le doute de faire leurs choix en connaissance de cause. Nous avons cependant constaté que des cas limites ne pouvaient être tranchés de façon certaine à la lumière des dispositions juridiques et de la jurisprudence actuelle. Ces situations pourront cependant s'apprécier au cas par cas, en appliquant judicieusement les principes légaux dont nous disposons.

Les analyses et développements proposés ont pu démontrer que les logiciels libres s'acclimataient sans trop de difficultés à nos règles de droit d'auteur. Nous pouvons de ce fait conclure positivement cette contribution en constatant que notre droit d'auteur peut également s'ériger en gardien des libertés.

PARTIE III

LOGICIELS LIBRES ET DROIT D'AUTEUR : LES DROITS MORAUX ET LES RÈGLES CONTRACTUELLES

Fabrice DE PATOUL

INTRODUCTION

157. La présente contribution abordera deux questions distinctes. Le premier chapitre examinera le sort des droits moraux en matière de logiciels libres (Chapitre 1). Le second chapitre analysera les règles contractuelles en matière de droit d'auteur dans le cadre des licences de logiciels libres (Chapitre 2).

La protection par le droit d'auteur des programmes d'ordinateurs octroie au développeur d'un logiciel des prérogatives patrimoniales et des prérogatives d'ordre moral. Les droits moraux ne font généralement pas l'objet de dispositions spécifiques dans les licences de logiciels libres. Cette situation s'explique notamment par l'influence du droit américain dans la rédaction des licences qui ne reconnaît aucun droit moral en matière de programmes d'ordinateurs. Chaque droit moral sera analysé individuellement et mis en perspective avec les licences de logiciel libre. Enfin, une dernière section abordera les limites des droits moraux à savoir la possibilité d'y renoncer ou de les exercer abusivement.

Les droits d'auteur peuvent faire l'objet d'une exploitation et de dispositions contractuelles. La législation belge sur le droit d'auteur établit des règles générales applicables à tous les contrats de licences ou de cession de droit d'auteur. Il conviendra d'envisager leur application aux licences des logiciels libres.

L'analyse se veut systématique. Il sera d'abord rappelé le cadre légal général du droit d'auteur, ensuite les dispositions spécifiques relatives aux programmes d'ordinateur pour enfin envisager leur application en matière de logiciel libre. Enfin, certaines licences seront étudiées de manière pragmatique.

CHAPITRE 1 : LE LOGICIEL LIBRE ET LES DROITS MORAUX

Section 1 : Considérations générales

§ 1. Les droits moraux

158. Les droits moraux protègent la personnalité de l’auteur dans son rapport à l’œuvre. Les droits moraux pourront être invoqués par l’auteur pour autant que l’atteinte en question ait un lien direct ou indirect avec son œuvre. Ainsi, une simple atteinte au nom de l’auteur sans rapport avec une de ses œuvres ne constitue pas une atteinte à son droit de paternité²⁴².

La loi belge générale sur le droit d’auteur (ci-après « LDA »)²⁴³ consacre trois prérogatives morales. Il s’agit des droits de divulgation, de paternité et d’intégrité (article 1^{er}, § 2 LDA) .

Les droits moraux sont réputés inaliénables (article 1^{er}, § 2, al. 1^{er} LDA). La loi ajoute que la renonciation globale à l’exercice futur des droits moraux est nulle. Elle autorise *a contrario* les renoncements qui ne présentent pas ces deux caractéristiques. Les droits moraux peuvent également être exercés de manière abusive. Ce détournement de l’exercice du droit moral pourra ainsi être sanctionné par l’application de la théorie de l’abus de droit.

En l’absence de disposition spécifique, la doctrine belge s’accorde à dire que la durée des droits moraux est identique à celle des droits patrimoniaux²⁴⁴. La durée des droits moraux est dès lors de septante ans après le décès de l’auteur (article 2, al. 1^{er} LDA).

²⁴² En revanche, l’auteur pourra éventuellement trouver une base légale grâce à ses droits de la personnalité. A. & H. J. LUCAS, *Traité de la propriété littéraire et artistique*, Paris, Litec, 2001, p. 303, n° 367 et s.

²⁴³ Loi du 30 juin 1994 relative au droit d’auteur et aux droits voisins, *M.B.*, 27 juillet 1994, p.19297-19314.

²⁴⁴ A. BERENBOOM, *Le nouveau droit d’auteur*, Bruxelles, Larcier, 1997, p. 238, n° 183 ; F. DE VISSCHER, B. MICHAUX, *Précis du droit d’auteur et des droits voisins*, Bruxelles, Bruylant, 2000, p. 147, n° 182 ; A. STROWEL, E. DERCLAYE, *Droit d’auteur et numérique : logiciels, bases de données, multimédia - Droit belge, européen et comparé*, Bruxelles, Bruylant, 2001, p. 88, n° 90. Pour mémoire, en France, le droit moral est considéré comme perpétuel.

La loi sur les programmes d'ordinateur (LPO) prévoit un régime particulier de droits moraux en faveur des développeurs de logiciels qu'il convient d'examiner.

§ 2. *Les droit moraux et les logiciels*

159. La loi sur les programmes d'ordinateur précise que « *le droit moral se règle conformément à l'article 6bis, 1. de la Convention de Berne* » (article 4 LPO)²⁴⁵. Le législateur belge a suivi la logique de la directive du 14 mai 1991 concernant la protection juridique des programmes d'ordinateur qui laissait aux Etats membres le soin de réglementer la matière des droits moraux, tout en imposant le respect de la Convention de Berne²⁴⁶.

Le texte de la Convention de Berne dispose qu'« *indépendamment des droits patrimoniaux d'auteur, et même après la cession desdits droits, l'auteur conserve pendant toute sa vie le droit de revendiquer la paternité de l'oeuvre et de s'opposer à toute déformation, mutilation ou autre modification de cette oeuvre ou à toute autre atteinte à la même oeuvre, préjudiciables à son honneur ou à sa réputation*» (article 6bis).

La Convention de Berne reconnaît uniquement deux droits moraux aux créateurs de programmes d'ordinateur : d'une part, celui de revendiquer la paternité de leur oeuvre, d'autre part, celui de s'opposer à toute atteinte à leur oeuvre préjudiciable à leur honneur et à leur réputation (droit à l'intégrité).

Le renvoi partiel à la Convention de Berne pose des difficultés par rapport à la situation des développeurs de logiciels, dans la mesure où il est par ailleurs admis que les questions non traitées par la loi spécifique sont soumises à la loi

²⁴⁵ Loi du 30 juin 1994 transposant la directive européenne du 14 mai 1991 sur la protection juridique des programmes d'ordinateur, *M.B.*, 27 juillet 1994, p. 19315-19317. Le législateur a préféré le terme « programme d'ordinateur » à celui de « logiciel ». Cette dernière notion est habituellement plus large dans la mesure où elle couvre le matériel de conception préparatoire et la documentation d'utilisation. S. DUSOLLIER, « La protection des programmes d'ordinateur », *Rép. Not.*, à paraître.

²⁴⁶ Directive 91/250/CEE du Conseil, du 14 mai 1991, concernant la protection juridique des programmes d'ordinateur, *J.O.C.E.*, n° L 122 du 17/05/1991 p. 0042 – 0046. Comme le souligne A. PUTTEMANS, les droits moraux sont « *un aspect toujours délicat des systèmes de droit d'auteur latino-germanique, d'une part, et du copyright anglo-saxon, de l'autre. Les Etats membres étaient donc libres de régir le droit moral comme ils l'entendaient, à la seule condition de ne pas manquer aux règles de la Convention de Berne* » (A. PUTTEMANS, « Au bout du droit d'auteur : la nouvelle protection juridique des programmes d'ordinateur », *R.D.C.*, 1995, p. 775).

sur le droit d’auteur²⁴⁷. On peut dès lors s’interroger sur l’existence ou non d’un droit de divulgation, consacré par la loi sur le droit d’auteur, mais pas mentionné dans la Convention de Berne.

La portée du droit à l’intégrité est également incertaine, car la solution minimaliste de la Convention de Berne est contredite par la loi sur le droit d’auteur qui reconnaît un droit absolu à l’intégrité de l’œuvre.

Les réponses à ces questions nécessitent de trouver l’articulation correcte entre trois textes : la loi générale sur le droit d’auteur, la loi spécifique sur la protection juridique des programmes d’ordinateurs et la Convention de Berne. L’étude de chacun des droits moraux nous donnera l’occasion d’analyser ces interactions.

§ 3. *Les droits moraux et les logiciels libres*

160. Les droits moraux revêtent un intérêt particulier dans le domaine des logiciels libres. Selon le modèle libre, l’auteur accorde à l’utilisateur des droits patrimoniaux extrêmement larges. En règle générale, l’utilisateur dispose du droit d’utiliser, reproduire, distribuer et modifier le logiciel.

L’auteur invoquera ses droits économiques uniquement dans la mesure où le preneur de licence ne respecte pas les termes de celle-ci. La marge de manœuvre de l’auteur se limitera donc à contrôler si la distribution et l’exploitation du logiciel est conforme au prescrit de la licence.

Si le développeur concède une licence sur l’ensemble de ses droits patrimoniaux, il conserve néanmoins l’intégralité de ses droits moraux. L’auteur d’un logiciel ou d’une partie de logiciel libre garde en effet le droit d’agir sur base de ses prérogatives morales.

A première vue, le régime des droits moraux semble mal s’accommoder avec les licences de logiciels libres qui permettent la libre modification du code du logiciel. Ainsi, ces licences ne prennent *a priori* pas en compte le droit à l’intégrité du développeur.

En revanche, elles accordent une place significative au droit de paternité. Les licences libres accordent une importance particulière à la mention du nom de l’auteur lors de chaque modification apportée au programme. Il est clair qu’un

²⁴⁷ F. BRISON, J.-P. TRIAILLE, « La nouvelle loi sur la protection des programmes d’ordinateur dans le sillage de la loi sur le droit d’auteur », *J. T.*, 1995, p. 142.

développeur ne retire souvent aucun avantage économique dans l'élaboration d'un logiciel libre. En revanche, il bénéficiera de la reconnaissance de ses pairs et pourra éventuellement valoriser économiquement sa réputation sur le marché des services gravitant autour des logiciels libres.

Chacun des droits moraux nécessite dès lors un examen particulier au regard des logiciels libres.

Section 2 : Le droit de divulgation

§ 1. Définition (art. 3, § 1^{er}, § 2, al. 3 et 4, LDA)

161. Le droit de divulgation permet à l'auteur de décider souverainement si son œuvre est achevée et quand et comment elle sera portée à la connaissance du public.

La notion de divulgation doit être comprise comme sa présentation au public en général et non à un tiers préalablement choisi (famille, ami, éditeur, etc.)²⁴⁸. Certains auteurs estiment que la divulgation comporte un élément matériel (l'acte de divulgation) et un élément moral (la volonté de l'auteur de divulguer son œuvre)²⁴⁹. Ainsi, une divulgation involontaire laisserait subsister le droit de divulgation.

L'auteur dispose également du droit de ne pas divulguer son œuvre même si elle est terminée. Le non-exercice du droit de divulgation a une incidence patrimoniale importante en ce qu'elle rend l'œuvre insaisissable (article 1^{er}, § 2, al. 4 LDA).

Selon la doctrine belge, le droit de divulgation est présenté habituellement comme définitif ou « irrévocable » dès qu'il est exercé par l'auteur. Ce droit s'épuise par la diffusion de l'œuvre par l'auteur auprès du public. En France, l'épuisement du droit de divulgation reste davantage controversé²⁵⁰.

²⁴⁸ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 82.

²⁴⁹ B. MICHAUX, F. DE VISSCHER, *op. cit.*, p. 151, n° 190 ; F. GOTZEN, « De algemene beginselen van de vermogensrechten en van de morele rechten van de auteur volgens de wet van 30 juni 1994 », in *Belgisch auteursrecht van oud naar nieuw – Le renouveau du droit d'auteur en Belgique*, Bruxelles, 1996, p. 82.

²⁵⁰ A. et H.-J. LUCAS, *Traité de la propriété littéraire et artistique*, Paris, Litec, 2001, p. 316, n° 386.

§ 2. Le droit de divulgation et les logiciels (art. 4, LPO)

162. L'existence d'un droit de divulgation en matière de programmes d'ordinateur est controversée en doctrine. Pour mémoire, en matière de droits moraux, la loi belge sur les programmes d'ordinateur renvoie au contenu de la Convention de Berne qui ne reconnaît pas le droit de divulgation. La base légale de ce droit semble ainsi *a priori* faire défaut²⁵¹.

Faut-il dès lors considérer que le programmeur ne bénéficie pas du droit de divulgation ?

Certains commentateurs de la loi belge partagent cet avis²⁵². Ils considèrent que le renvoi exprès à la Convention de Berne exclut le droit de divulgation et déroge dès lors au droit commun. En outre, les travaux préparatoires précisent que le législateur entend ne mettre en place qu'« un droit moral minimal »²⁵³.

Toutefois, une partie de la doctrine, que nous rejoignons, considère que le droit de divulgation de la loi générale sur le droit d'auteur est d'application²⁵⁴. Cette opinion s'appuie sur la directive du 14 mai 1991 qui impose une assimilation des programmes d'ordinateurs aux œuvres littéraires. Or, l'article 1^{er}, § 2 LDA consacre un droit de divulgation à l'auteur d'une œuvre littéraire. Cette interprétation repose aussi sur l'exposé des motifs de la loi sur la LPO qui précise que « le droit de divulgation se règle conformément au droit commun »²⁵⁵. Enfin, l'expression de droit moral « minimum » utilisée dans les travaux préparatoires ne semble se rapporter qu'au droit à l'intégrité et non à l'ensemble des droits moraux (*infra*, n° 170 et s.). Par conséquent, rien ne permet d'affirmer avec certitude que le législateur ait souhaité priver les créateurs de programmes d'ordinateurs du droit de divulgation. Les exceptions au droit commun étant de stricte interprétation, il convient de privilégier le

²⁵¹ S. DUSOLLIER, « La protection des programmes d'ordinateur », *Rép. Not.*, à paraître.

²⁵² A. PUTTEMANS, *op. cit.*, p. 775 ; A. STROWEL, « La loi belge du 30 juin 1994 sur les programmes d'ordinateur : vers un droit d'auteur *sui generis* ? », *R. I. D. A.*, 1995, n° 164, p. 201 ; A. STROWEL, J.-P. TRIAILLE, *Le droit d'auteur : du logiciel au multimédia*, Diegem, Kluwer Editions Juridiques, 1997, p. 165-166 ; A. BERENBOOM, *op. cit.*, p. 225, 172bis.

²⁵³ Rapport au nom de la Commission de la justice par M. DE CLERCK, *Doc. Parl.*, Ch. Repr., n°1071/6, sess. ord., 1993-1994, p. 11-14, du 17 mars 1994 ; Rapport au nom de la Commission de la justice par M. ERDMAN, *Doc. Parl.*, Sén., sess. ord. 1991-1992, n°1054-2, p. 6.

²⁵⁴ B. MICHAUX, F. DE VISSCHER, *op. cit.*, p. 229, n° 269. ; F. BRISON, J.-P. TRIAILLE, « La nouvelle loi sur la protection des programmes d'ordinateur dans le sillage de la loi sur le droit d'auteur », *J. T.*, 1995, p. 142. Pour une position nuancée : A. STROWEL et E. DERCLAYE, *op. cit.*, p. 211, n° 238.

²⁵⁵ Rapport DE CLERCK, *op. cit.*, p. 3.

régime de droit commun de la LDA en cas de doute sur la volonté du législateur.

Comme le soulignent A. STROWEL et E. DERCLAYE, le droit de divulgation est susceptible de jouer un rôle important en pratique²⁵⁶. En effet, de la divulgation dépend le caractère saisissable ou non du programme informatique. Ainsi, si un logiciel est développé « en interne » par une société informatique qui tombe ensuite en faillite, ce logiciel ne pourra être saisi par les créanciers de la société s'il n'a pas été préalablement divulgué au public.

§ 3. *Le droit de divulgation et les logiciels libres*

163. Internet constitue le moyen de communication privilégié des concepteurs de logiciels libres. La mise à disposition d'un logiciel libre sur un site web, un babilleur électronique (ou « *bulletin board system* ») ou tout autre moyen le rendant accessible au public constitue une divulgation du logiciel. Dans cette hypothèse, le droit du concepteur de divulguer son programme ou son apport à celui-ci sera en principe épuisé.

Une licence de logiciel libre ne peut imposer à un développeur de diffuser les modifications qu'il apporte au logiciel. Une telle obligation priverait en effet le développeur du droit ne pas divulguer son logiciel. En vertu du principe d'inaliénabilité du droit moral et de l'interdiction de toute renonciation globale et future, une clause formulée de manière générale qui exigerait la divulgation de toute modification apportée au logiciel serait, à notre avis, nulle²⁵⁷.

Le mode de conception des logiciels libres ne permet pas toujours d'établir quand, comment et par qui la divulgation se réalise. Un logiciel libre est souvent développé par une communauté de programmeurs où chacun propose et discute les améliorations possibles. Ce logiciel en cours de conception, sorte de « *work in progress* », est généralement accessible et soumis à l'évaluation des seuls participants au projet et non du grand public. Différentes versions du programme vont ainsi se succéder et être testées avant d'aboutir à une version stable et en état de fonctionnement. Seule cette dernière version sera mise à disposition du public.

Selon nous, cette mise à la disposition du logiciel à la communauté et au public en général sera la seule qui épuisera le droit de divulgation de chacun des auteurs, pour autant que chacun d'eux ait marqué son accord tacite ou exprès sur cette publication.

²⁵⁶ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 210, n° 238.

²⁵⁷ Sur le principe d'inaliénabilité, voyez *infra* n° 175.

A défaut, un des concepteurs du logiciel pourrait théoriquement s'y opposer sur la base de son droit exclusif de divulgation. Rappelons que l'exercice du droit de divulgation est limité par la théorie de l'abus de droit (*infra*, n° 178 et s.). Le programmeur exercerait abusivement son droit de divulgation si son but était notamment de porter préjudice à autrui, de causer un dommage sans intérêt pour lui-même ou de s'octroyer un avantage économique. Tel serait le cas d'un programmeur qui attendrait le dernier moment pour poser son veto à la sortie du logiciel malgré l'accord de l'ensemble des autres programmeurs ayant participé à sa création.

Vu le nombre élevé de personnes participant à la création d'un logiciel libre, il existe des risques de désaccord sur l'état d'avancement du logiciel et sa date éventuelle de sortie. En cas de conception collective d'un logiciel libre, des solutions juridiques pourraient être mises en oeuvre pour éviter d'éventuels conflits liés à l'exercice du droit de divulgation.

164. Des aménagements contractuels sont envisageables. Le responsable ou le gérant d'un projet pourrait demander aux membres de respecter un code de bonne conduite ou une charte avec des conditions générales. Dans ce cadre, la divulgation du logiciel pourrait être modalisée contractuellement ou organisée selon une procédure précise. Ainsi, tout concepteur de logiciel pourrait être consulté préalablement à la publication d'un logiciel et disposerait d'un délai pour s'y opposer en indiquant les motifs d'un éventuel refus de publication.

Cette simple modalisation de l'exercice du droit de divulgation permettrait de faciliter la preuve d'un éventuel abus de droit. Néanmoins, ce type de clause ne pourrait-il pas être analysé comme une renonciation globale et future au droit de divulgation prohibée en droit d'auteur ? Ces clauses seraient à notre avis valables si le programmeur ne renonce pas à l'exercice de son droit au moment de la conclusion du contrat ou de manière absolue.

Par ailleurs, les membres ayant conçu le logiciel pourraient également s'engager à accepter la divulgation du logiciel dès qu'il remplit certaines caractéristiques et fonctions préalablement définies²⁵⁸.

Enfin, des accords ou des clauses de confidentialité sont également envisageables en matière de logiciels libres. Ainsi, un développeur de logiciel libre peut s'engager à ne pas diffuser les modifications avant la sortie « officielle » du logiciel décidée par le gérant de la communauté. Par ailleurs, une société peut également faire signer à un développeur une clause de confidentialité par laquelle il accepte de ne pas communiquer les adaptations

²⁵⁸ F. BRISON, B. MICHAUX, « De nieuwe auteurswet (2de deel) », *R.W.*, n° 59/16, 1995, p. 524-525; B. MICHAUX, F. DE VISSCHER, *op. cit.*, p. 153, n° 194.

réalisées sur un logiciel libre pour les besoins de sa société. Cela ne dispense toutefois pas la société de respecter les termes de la licence libre qui, le cas échéant, l'obligera à divulguer une modification si elle souhaite commercialiser la nouvelle version du logiciel.

§ 4. *Le droit de divulgation et les licences de logiciels libres*

165. Les licences de logiciels analysées ne prennent pas en compte le droit de divulgation. Ces licences imposent que les adaptations apportées aux logiciels libres soient diffusées dans les mêmes conditions que la licence.

Ces licences de logiciels libres n'imposent pas aux développeurs de divulguer les modifications apportées aux logiciels. Un particulier ou une société ont ainsi le droit de modifier un logiciel libre et de l'utiliser de manière privée ou de manière interne sans jamais diffuser cette version modifiée du logiciel. Selon certaines licences « copyleft », en cas de diffusion de cette nouvelle version dans le public, les codes sources du logiciel doivent néanmoins être communiqués et diffusés sous la licence appropriée.

C. CARON s'inquiète d'une possible « paralysie du système » dans l'hypothèse où un développeur ne diffuse pas la modification qu'il a apportée à un logiciel²⁵⁹. Cette crainte nous paraît injustifiée dans la mesure où souvent plusieurs versions du logiciel circulent simultanément et font l'objet de modifications. L'absence de contribution d'un développeur sera rapidement compensée par celle d'un autre qui travaille en parallèle sur une autre version du logiciel.

Certaines des licences libres analysées imposent des conditions en cas de divulgation d'un programme. Par exemple, l'obligation de diffuser le logiciel avec une licence identique (ex. la G.P.L.). Certains commentateurs français estiment que ces conditions sont susceptibles de constituer des atteintes au droit de divulgation dans la mesure où elles limitent les modalités de divulgation de l'auteur²⁶⁰. Rappelons qu'en France, le caractère inaliénable des droits moraux est considéré comme étant d'ordre public. On ne peut renoncer ou poser des conditions à l'exercice des droits moraux. En revanche, le législateur belge a uniquement interdit les renonciations globales à l'exercice futur du droit. Des renonciations partielles au droit de divulgation sont par conséquent autorisées.

²⁵⁹ C. CARON, « Les licences de logiciels dits « libres » à l'épreuve du droit d'auteur français », *D.*, 2003, p. 1559.

²⁶⁰ C. ROJINSKY, V. GRYNBAUM, « Les licences libres et le droit français », *Propriétés Intellectuelles*, Juillet 2002, n° 4, p. 36.

Le développeur est tenu de respecter la licence qui accompagne le logiciel. Par conséquent, il accepte implicitement que l'exercice de son droit de divulgation soit soumis aux termes de la licence, à l'instar de ses droits patrimoniaux.

Bien que la plupart des licences ne le mentionnent pas toujours clairement, seuls les apports dérivés du logiciel sont soumis aux termes de la licence. S'il s'agit d'une œuvre indépendante et autonome, le programmeur demeure libre de déterminer les conditions de divulgation et de diffusion du logiciel (article 2§2 G.P.L.).

Les licences libres analysées demeurent souvent très souples et peuvent être combinées avec des accords de confidentialité.

Prenons l'exemple de la G.P.L.

Une société peut demander à un développeur d'adapter un logiciel libre à ses besoins et de garder confidentielles les modifications apportées. Dans cette hypothèse, les termes de la G.P.L. seront respectés dans la mesure où la société conserve la possibilité de distribuer le logiciel. En pratique, elle n'aura souvent aucun intérêt à exercer cette faculté.

Par contre, un développeur ne pourra pas commercialiser une version modifiée d'un logiciel sous G.P.L., en interdisant à son client toute distribution du logiciel. En effet, dans ce cas, le développeur ne respecte pas les termes de la G.P.L. dans la mesure où la société ne dispose plus du droit de distribuer cette version amendée du logiciel.

Section 3 : Le droit de paternité ou d'attribution

§ 1. Définition (art. 1^{er}, § 2, al. 5, LDA)

166. Le droit de paternité permet à l'auteur de revendiquer ou de refuser la paternité de l'œuvre.

En vertu du droit de paternité, l'auteur peut exiger que son nom figure sur les exemplaires de l'œuvre ou, au contraire, souhaiter l'anonymat ou encore l'utilisation d'un pseudonyme. Le droit de paternité permet également à l'auteur de s'opposer à tout tiers qui tenterait d'usurper la qualité d'auteur.

L'exercice du droit de paternité présente également un intérêt pratique non négligeable. La mention de la signature de l'auteur sur une œuvre ou d'un signe permettant de l'identifier constitue une présomption légale de paternité (article

6, al. 2 LDA). En d'autres termes, la personne physique ainsi identifiée est réputée être l'auteur de l'œuvre.

Des tensions peuvent naître entre l'auteur et le propriétaire matériel du support de l'œuvre. Le droit de l'auteur d'exiger l'apposition de son nom peut entrer en conflit avec le droit de propriété de celui qui a acquis l'objet incorporant l'œuvre.

Est-ce que l'auteur peut toujours en toutes circonstances exiger que son nom figure sur le support incorporant l'œuvre ? Sur base d'une jurisprudence ancienne de la Cour de cassation, certains commentateurs considèrent que le droit de paternité ne contient pas un « droit positif de signature » en faveur de l'auteur qui l'autoriserait à imposer la mention de son nom au propriétaire de l'objet incorporant son œuvre²⁶¹. D'autres auteurs sont moins catégoriques ou appellent à un examen des modalités contractuelles d'exécution de l'œuvre²⁶². Ces auteurs renvoient donc implicitement à une analyse de la commune intention des parties. Une solution qui privilégie le droit de propriété ou le droit de paternité sera toujours délicate dans la mesure où chaque intervenant s'appuie sur des prérogatives légitimes.

§ 2. *Le droit de paternité et les logiciels (art . 4 LPO)*

167. La Convention de Berne à laquelle le législateur belge renvoie, consacre un droit de paternité en faveur du programmeur.

Le créateur du logiciel se voit reconnaître le droit de revendiquer la paternité de son programme d'ordinateur. En pratique, les logiciels sont souvent développés par des sociétés où travaillent des équipes d'informaticiens. Il est dès lors difficile de mentionner les noms de tous les « auteurs » du programme. Un nouveau logiciel est ainsi commercialisé comme le produit d'une société et non d'un ou plusieurs développeurs particuliers²⁶³.

Certains logiciels continuent néanmoins à être développés par des programmeurs indépendants. Dans ce cas, le droit de paternité s'exercera

²⁶¹ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 83, n° 82 ; Cass., 16 janvier 1941, *Pas.*, 1941, I, p. 11 : la mention du nom de l'architecte sur un édifice requiert le consentement de son propriétaire.

²⁶² A. BERENBOOM, *op. cit.*, p. 140, n° 100 ; F. DE VISSCHER et B. MICHAUX, *op. cit.*, p. 155, n° 195.

²⁶³ Certains logiciels particuliers mentionnent parfois l'ensemble des développeurs du programme. Tel est le cas notamment des jeux informatiques où les noms de tous les créateurs du logiciel sont indiqués.

généralement par une mention du nom de l'auteur sous la forme de lignes de code dans le langage de programmation, ou dans la licence accompagnant le logiciel ou dans une fenêtre lors de l'installation ou de l'exploitation du logiciel.

Dans le monde informatique, les développeurs affectionnent l'emploi de pseudonymes sous forme de surnoms (ou en anglais « nicknames »). L'usage du pseudonyme est un mode d'exercice du droit de paternité autorisé par la loi²⁶⁴. Le recours à un pseudonyme peut toutefois avoir une influence sur la durée de protection. En effet, si le pseudonyme ne permet pas de déterminer avec suffisamment de certitude l'identité du programmeur, la durée du droit d'auteur sera de septante ans à partir du moment où le programme est licitement rendu accessible au public (article 2, § 1, al. 2 et 3 LDA).

§ 3. Le droit de paternité et les logiciels libres

168. Comme indiqué précédemment, le droit de paternité acquiert une importance particulière dans le cadre des logiciels libres.

Différentes raisons peuvent expliquer ce regain d'intérêt pour le droit de paternité. Un logiciel libre est souvent développé par des informaticiens indépendants et non une société spécifique. L'auteur d'un logiciel libre accorde dès lors de l'importance à être reconnu comme l'auteur ou co-auteur de celui-ci dans la mesure où le rapport à son œuvre est plus direct et personnel.

Le développeur d'un logiciel libre revendiquera sa qualité d'auteur et l'originalité de son apport dans le logiciel. Dans l'univers plus fonctionnel et économique de l'informatique, on assiste à un renouveau de la figure classique de l'auteur qui est à la source de l'œuvre et qui marque celle-ci de son « empreinte personnelle »²⁶⁵.

Si l'auteur est prêt à « sacrifier » certaines de ses prérogatives patrimoniales, il entend conserver et valoriser ses droits moraux dont celui d'être présenté comme le créateur de l'œuvre. Cet exercice du droit moral de paternité n'est toutefois pas complètement désintéressé. En effet, le marché croissant des services dérivés entourant l'utilisation des logiciels libres constitue une opportunité commerciale pour les développeurs de logiciels libres. Le fait de pouvoir se présenter comme l'auteur d'un logiciel constitue un avantage

²⁶⁴ V. CASTILLE, « Les droits moraux et la bande dessinée », in *Le droit d'auteur et la bande dessinée*, Bruylant, 1997, p. 272.

²⁶⁵ Pour une réflexion plus philosophique sur la figure de l'auteur dans le logiciel libre, voyez S. DUSOLLIER, « *Open source* et *Copyleft* : une remise en cause de la figure de l'auteur ? », in A. STROWEL (éd.) *Autour de la figure de l'auteur*, Bruxelles, Bruylant, à paraître.

concurrentiel non négligeable sur d'autres informaticiens ou sociétés informatiques.

§ 4. Le droit de paternité et les licences de logiciels libres

169. La licence G.P.L. souligne à plusieurs reprises l'importance de la mention du droit d'auteur (« *copyright notice* ») lors de l'exploitation des logiciels²⁶⁶. Chaque copie du logiciel distribuée sous la licence G.P.L. doit être accompagnée de la notice (article 1^{er} G.P.L.). En outre, la G.P.L. impose à tous les utilisateurs de conserver toutes les précédentes notices de droit d'auteur qui se réfèrent à la licence. Il s'agit en réalité de la seule partie du logiciel que l'utilisateur ne pourra modifier. Selon le principe du modèle libre, toute autre modification est autorisée pour autant qu'elle soit clairement indiquée et datée par l'utilisateur (article 2, a) G.P.L.). En cas de programmes utilisant des commandes interactives, la licence impose qu'une annonce avec la notice de droit d'auteur s'affiche automatiquement lors de la mise en œuvre du logiciel (article 2, c) G.P.L.).

La licence G.P.L. ne précise pas le contenu de cette notice sinon qu'elle doit être « appropriée » (article 1^{er} G.P.L.).

Le site web www.gnu.org, qui héberge les licences G.P.L., propose des modalités pratiques afin d'appliquer la licence à un programme informatique²⁶⁷. Des modèles de notices sont proposés reprenant la mention classique « Copyright © année nom de l'auteur ». Il est ensuite conseillé de rappeler l'identité et les coordonnées du programmeur, les fonctionnalités du logiciel et les principes de la G.P.L.

Les rédacteurs de la licence précisent également que les auteurs des modifications indiquent leur nom afin de distinguer les différentes versions du logiciel et protéger la réputation des autres auteurs²⁶⁸. Une telle obligation risque d'être en contradiction avec le droit dont disposent les développeurs de se voir ou non attribuer leur contribution au logiciel.

²⁶⁶ En ce qui concerne le droit de paternité, la L.G.P.L. contient les mêmes types de dispositions que la G.P.L. Les commentaires de la licence G.P.L. peuvent dès lors être transposés à la L.G.P.L.

²⁶⁷ Voyez la section « How to apply these terms to your new programs » sur le site <http://www.gnu.org/licenses/gpl.html>.

²⁶⁸ Voyez la foire aux questions accessible à l'adresse <http://www.gnu.org/licenses/gpl-faq.html>.

L'O.S.L. traite expressément la question du droit de paternité en son article 6 intitulé « *attribution rights* ». Cette licence oblige l'utilisateur à conserver toutes les notices de droit d'auteur, de brevets ou de droit des marques comprises dans la licence lors de la distribution de chaque copie du logiciel ou d'une de ses versions dérivées.

L'O.S.L. impose également que toute modification du logiciel par l'utilisateur soit expressément mentionnée par une notice d'attribution appropriée. La notice semble devoir contenir l'existence de la modification et l'identité de son auteur (« (...) *to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work* »).

Les licences B.S.D. et Q.P.L. prévoient également l'obligation de conserver les notices de droit d'auteur en cas de copies ou de modifications du logiciel. Ces licences ne précisent pas si les auteurs des modifications doivent être indiqués, ni de quelle manière.

L'imposition d'une signature constitue à notre avis une violation du droit de paternité qui permet à l'auteur de refuser l'attribution d'une œuvre²⁶⁹. Par conséquent, une clause qui impose une obligation positive au programmeur de mentionner son nom sur son œuvre est nulle.

L'analyse des différentes licences démontre que le respect du droit de paternité est souvent pris en compte dans les licences de logiciels libres. L'écueil à éviter est de vouloir imposer la mention de leur identité aux auteurs des modifications.

Section 4 : Le droit à l'intégrité ou au respect de l'œuvre

§ 1. Définition (art. 1^{er}, § 2, al. 6 et 7, LDA)

170. Le droit à l'intégrité ou au respect de l'œuvre permet à l'auteur de s'opposer à toute modification de son œuvre (article 1^{er}, § 2, al. 6, LDA).

La loi précise que l'auteur conserve son droit de s'opposer à toute déformation, mutilation ou autre modification de son œuvre préjudiciable à son honneur ou à sa réputation même en cas de renonciation à son droit moral (article 1^{er}, § 2, al. 7, LDA).

²⁶⁹ Du même avis, A. METSGER, T. JAEGER, « Open source Software and German Copyright Law », *IIC*, vol. 32, 2001/1, p. 66.

On distingue dès lors habituellement deux niveaux au droit à l'intégrité.

Le premier niveau permet à l'auteur de s'opposer à toute modification de son œuvre sans qu'il ait à démontrer un quelconque préjudice dans son chef. L'auteur pourra valablement renoncer à exercer ce droit à l'égard de certaines modifications préalablement définies ou raisonnablement déterminables²⁷⁰.

Un second niveau au droit à l'intégrité est plus fondamental et subsiste même en cas de renonciation par l'auteur à l'exercice de son droit. La loi permet à l'auteur de s'opposer à toute atteinte à son œuvre qui serait préjudiciable à son honneur ou à sa réputation.

En principe, ce droit au respect plus restreint est soumis au respect de deux conditions cumulatives :

- une modification au sens large de l'œuvre (« *déformation, mutilation ou autre modification de cette oeuvre ou à toute autre atteinte à la même œuvre* »)
- préjudiciable à l'honneur ou à la réputation de l'auteur

Toutefois, la définition large de la notion de modification (« *toute atteinte* ») facilite la réalisation de la première condition. Ainsi, un simple changement d'environnement ou une atteinte à l'esprit de l'œuvre peut suffire²⁷¹. C'est essentiellement le préjudice à l'honneur ou à la réputation de l'auteur qu'il faudra établir.

§ 2. Le droit à l'intégrité et les logiciels (art. 4 LPO)

171. Le droit à l'intégrité du concepteur d'un logiciel se limite au second niveau du droit à l'intégrité, qui suppose l'établissement d'un préjudice à

²⁷⁰ La doctrine diverge sur la portée de la notion de « modification » reprise à l'al. 6, A. STROWEL et E. DERCLAYE considèrent qu'il s'agit uniquement de modifications relatives à l'objet de l'œuvre. Par conséquent, un simple changement de l'environnement ne sera pas considéré comme une atteinte sauf si la preuve d'un préjudice à l'honneur ou à la réputation exigée au dernier alinéa de l'article 1^{er}, § 2, LDA est apportée (*op. cit.*, p. 84, n° 83 et note 276). En revanche, F. DE VISSCHER, B. MICHAUX soulignent l'étendue du terme « modification » qui permet à l'auteur de s'opposer à toute atteinte même ne portant pas sur l'objet lui-même (*op. cit.*, p. 157, n° 200 et note 375).

²⁷¹ A. STROWEL et E. DERCLAYE, *op. cit.*, p. 84, n° 83. Les auteurs donnent l'exemple du déplacement d'une sculpture conçue pour s'intégrer à un ensemble architectural donné. Voyez également F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 157, n° 200 et s. Voy. égal. C. CARREAU, « Droit d'auteur et esprit de l'œuvre », *RIDA*, Juillet 2004.

l'honneur ou à la réputation. Cette limitation du droit à l'intégrité résulte du renvoi à l'article 6 *bis* de la Convention de Berne par le législateur belge²⁷².

En pratique, le préjudice à l'honneur ou à la réputation d'un concepteur d'un logiciel s'avère difficile à démontrer. En effet, le caractère habituellement fonctionnel d'un programme d'ordinateur, rend malaisé l'appréciation d'une atteinte à l'honneur ou à la réputation. Tel pourrait être le cas lors de la traduction d'un programme dans un autre langage ou de modifications à un logiciel qui rendraient le programme moins performant ou en y introduisant des bugs qui nuiraient à la réputation de l'auteur²⁷³.

§ 3. *Le droit à l'intégrité et les logiciels libres*

172. Le logiciel distribué sous forme de licence libre a, par nature, vocation à être modifié à de nombreuses reprises. En effet, le concepteur du logiciel autorise sous forme de licence la modification du logiciel par les tiers. Si ces stipulations contractuelles permettent une modification au point de vue des droits patrimoniaux, elles laissent intacts les droits moraux.

Est-ce que le concepteur d'un logiciel libre pourrait dès lors invoquer son droit à l'intégrité à l'encontre d'un autre programmeur ayant modifié le logiciel de telle manière qu'il porte atteinte à sa réputation ?

En principe, le concepteur d'un logiciel qui le distribue sous licence libre conserve son droit à l'intégrité. Par conséquent, quel que soit le prescrit de la licence, le développeur d'un logiciel libre est toujours en mesure de s'opposer à une modification de son logiciel qui porterait atteinte à son honneur ou à sa réputation.

Prenons l'exemple d'un logiciel sans défaut connu qui est distribué sous licence libre. Est-ce que le concepteur initial peut agir contre un développeur qui aurait introduit des bogues dans son logiciel ?

En pratique, le concepteur est tenu d'établir le préjudice à sa réputation ou à son honneur. On connaît l'importance que revêt la réputation pour un développeur de logiciel libre qui ne dispose plus de ses prérogatives patrimoniales. Ses qualités de programmeur constituent sa « carte de visite » pour proposer ses services. L'association de son nom avec un programme défaillant pourrait, à notre avis, lui être préjudiciable et lui donner une base pour agir en justice.

²⁷² Le Code français de la propriété intellectuelle L. 121-7 adopte une solution identique.

²⁷³ A. PUTTEMANS donne l'exemple de la modification d'un jeu pour enfants en programme faisant l'apologie du nazisme. A. PUTTEMANS, *op. cit.*, p. 775, note 34.

D'autres cas de figures sont envisageables. En effet, l'altération de l'objet même du programme n'est pas exigée pour permettre au programmeur d'invoquer son droit moral au respect. Le simple changement de contexte du logiciel ou une dénaturation de l'«esprit» de celui-ci peut constituer une atteinte à l'honneur ou à la réputation de son auteur.

Que penser de l'utilisation d'un logiciel libre par une grande société informatique adepte du modèle propriétaire qui ne recueille pas les sympathies du développeur ? Ou par un fabricant d'armes, une entreprise peu scrupuleuse de l'environnement ou l'administration d'un pays dictatorial ? Les exemples peuvent être multipliés à l'infini.

La question centrale est de déterminer ce que l'on entend par une atteinte à l'honneur ou à la réputation relativement à un programme informatique. Il incombera au programmeur d'établir l'existence d'une telle atteinte.

Selon nous, l'objectif des programmeurs libres est davantage de développer des solutions informatiques stables et perfectibles grâce à l'accès permanent aux codes sources. Les utilisateurs des logiciels libres ont d'abord l'obligation de respecter les termes de la licence. Hormis les cas d'atteinte flagrantes à l'honneur et à la réputation, la libre modification et distribution conférées par le concepteur d'un logiciel libre ne lui permettent pas de s'opposer, par une appréciation subjective et discrétionnaire, à certaines utilisations ou à certains utilisateurs particuliers. Si tel est son souhait, le programmeur devrait le mentionner clairement et préalablement dans une clause appropriée de sa licence. Le programmeur pourra ensuite invoquer la violation de cette clause contre un tiers qui ferait une utilisation prohibée du logiciel.

En définitive, si le développeur d'un logiciel libre conserve son droit moral à l'intégrité, il ne réussira à l'invoquer, à notre avis, que de manière exceptionnelle à moins de préciser davantage les utilisations autorisées ou interdites du logiciel ce qui n'est pas souvent conciliable avec la libre utilisation exigée par les licences libres.

§ 4. Le droit à l'intégrité et les licences de logiciels libres

173. Les licences de logiciels libres analysées autorisent les licenciés à modifier le logiciel. Ainsi, la G.P.L. indique par exemple « *Vous pouvez modifier votre ou vos copies du Logiciel ou toute partie du Programme, c'est-à-dire réaliser une œuvre basée sur le logiciel, et copier et distribuer ces*

modifications ou cette œuvre (...) ²⁷⁴» (article 2 G.P.L.). Les seules conditions qui sont habituellement posées par la licence sont l'apposition d'une notice de droit d'auteur.

Or, les clauses relatives aux œuvres traditionnelles permettant d'apporter des modifications à une œuvre seront nulles si elles sont rédigées de manière générale²⁷⁵. Cette règle se déduit *a contrario* du droit inaliénable de l'auteur de pouvoir toujours s'élever contre une atteinte qui serait préjudiciable à son honneur ou à sa réputation. En effet, si l'auteur autorise toute modification de son logiciel, il renonce de manière globale à pouvoir exercer son droit à l'intégrité.

Selon une interprétation stricte de la loi, les clauses des licences de logiciels prévoyant une telle possibilité sont nulles.

Toutefois, ces clauses nous semblent davantage centrées sur les droits patrimoniaux et ne visent pas à priver le donneur de licence de son droit à l'intégrité. Selon nous, la particularité des licences de logiciels libres est de permettre une large possibilité de modifications. En d'autres termes, le donneur de licence qui invoque son droit à l'intégrité ne pourra se voir opposer le droit de modification qu'il a concédé via la licence.

Par ailleurs, l'obligation d'exécuter de bonne foi les conventions permet également d'encadrer l'exercice des droits par les parties (art. 1134, al. 3, C. civ.). L'imposition de l'exécution de bonne foi de la convention interdit aux parties d'abuser des prérogatives qu'elles puisent dans le contrat²⁷⁶. Ainsi, le donneur de licence et le licencié ne pourront invoquer respectivement le droit à l'intégrité et le droit de modification de manière abusive²⁷⁷.

Les licences reprises dans notre analyse n'abordent pas la question du droit moral à l'intégrité de l'auteur. La plupart des licences se contentent d'obliger le développeur à mentionner les modifications et leurs dates (articles 2, a) G.P.L. et

²⁷⁴ Traduction libre.

²⁷⁵ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 91, n° 161 *in fine*; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 148, n° 183 ; A. CRUQUENAIRE, E. DERCLAYE, « Quelques considérations sur les modalités d'intervention en justice des sociétés collectives, sur la portée de certaines exceptions en droit d'auteur, et sur l'évaluation du préjudice résultant d'une atteinte au droit d'auteur », note sous Bruxelles (9^e ch.), 23 mars 2001, *A.&M.*, 2001, p. 379.

²⁷⁶ J-L. FAGNART, « L'exécution de bonne foi des conventions : un principe en expansion », *R.C.J.B.*, 1986, p. 316.

²⁷⁷ A. & H. J. LUCAS, *Traité de la propriété littéraire et artistique*, Paris, Litec, 2001, p. 344, n° 325.

2 b) L.G.P.L. et article 6 O.S.L.). Ces mentions seront éventuellement utiles pour identifier l'auteur d'une modification préjudiciable.

Les préambules de la G.P.L. et la L.G.P.L. préviennent néanmoins l'utilisateur en ces termes « si le logiciel est modifié par un tiers et transmis, nous voulons que les destinataires sachent que ce qu'ils détiennent n'est pas l'original, et donc que tout dysfonctionnement introduit par d'autres ne reflète pas la réputation des auteurs originaires »²⁷⁸.

Cette disposition de la G.P.L. laisse perplexe dans la mesure où elle ne contient ni obligation, ni affirmation d'un droit²⁷⁹.

Du point de vue de l'utilisateur du logiciel, il s'agit d'un simple avertissement que le logiciel est susceptible de contenir des bogues qui ne sont pas le fruit des développeurs initiaux du programme.

Du point de vue du développeur, cette disposition pourrait s'interpréter comme une renonciation tacite du développeur à exercer son droit au respect de son œuvre. En effet, le développeur pourra difficilement se prévaloir de son droit à l'intégrité après avoir averti que les éventuels bogues du logiciel ne pouvaient lui être imputés ou nuire à sa réputation.

Toutefois, en application du principe de l'inaliénabilité des droits moraux, l'auteur ne peut renoncer globalement à son droit à l'intégrité. Si cette disposition venait à être assimilée à une renonciation globale et future, elle serait nulle²⁸⁰.

Cette disposition doit plutôt être analysée comme une simple mise en garde de l'utilisateur sans véritable portée juridique, d'où sa place dans le préambule plutôt que dans le corps de la licence. En effet, elle souligne les limites légales entourant le droit de modification. Tout développeur est tenu d'apporter des modifications qui ne portent pas atteinte à la réputation et à l'honneur des autres développeurs ayant contribué à la conception du logiciel. Il s'agit d'un simple rappel du minimum légal de la Convention de Berne.

²⁷⁸ Traduction libre.

²⁷⁹ M. CLEMENT-FONTAINE voit dans cette disposition du préambule une obligation qui doit être interprétée comme une application fidèle de l'article 121-7, 1°, du Code de propriété intellectuelle français qui consacre le droit à l'intégrité en matière de logiciel. Cette interprétation nous semble audacieuse par rapport au texte de la disposition.

²⁸⁰ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 91, n° 161. En ce qui concerne, la renonciation au droit moral, voyez *infra*, n° 174 et s.

Les licences de logiciels libres gagneraient à une plus grande clarté dans leur rédaction en rappelant que tout usage du logiciel reste soumis au respect de l'honneur ou de la réputation de son ou ses auteur(s). Si le programmeur souhaite réserver des usages particuliers ou des utilisateurs particuliers, il devrait également l'indiquer dans les termes de la licence.

Dans le même ordre d'idées, les licences pourraient mentionner une réserve dans la clause de cession des droits en précisant que « *toute cession n'emporte pas le droit pour le développeur d'exercer son droit de modification qui porterait atteinte à l'honneur ou à la réputation* ».

Section 5 : Les limites des droits moraux

174. La loi octroie des prérogatives morales exclusives aux auteurs. Il en découle que l'autorisation préalable de l'auteur est en principe requise. L'exercice du droit moral n'est toutefois pas absolu. L'exercice d'un droit moral peut être considéré comme abusif (§ 2). Par ailleurs, il convient également de rappeler que l'auteur peut, sous certaines conditions, renoncer à son droit moral. Cette restriction au droit moral nous donne l'occasion d'examiner la question du caractère inaliénable des droits moraux en matière de programmes informatiques (§ 1).

§ 1. La renonciation au droit moral

A. Principe (art. 1^{er}, § 2, al. 1 et 2, LDA)

175. Selon la LDA, les droits moraux sont inaliénables (article 1^{er}, § 2, al. 1 LDA). En d'autres termes, les droits moraux ne peuvent être cédés à titre onéreux ou à titre gratuit.

La loi précise également que la renonciation globale à l'exercice futur du droit moral est nulle (article 1^{er}, § 2, al. 1 LDA). Une telle renonciation aurait le même effet qu'une aliénation car elle priverait l'auteur de l'exercice de ses droits moraux. Par conséquent, le législateur a voulu éviter que la règle d'inaliénabilité soit vidée de sa substance par le biais d'une renonciation pure et simple de l'auteur à l'ensemble de ses droits.

En réalité, seules les renonciations globales et qui concernent l'exercice futur du droit moral sont interdites. La loi admet *a contrario* les renonciations partielles (et non « globales ») ou les renonciations à l'égard d'atteintes déjà portées (et

non « futures »). Dans ces hypothèses, on considère que l'auteur renonce à ses droits en connaissance de cause²⁸¹.

Une renonciation à un droit moral doit-elle être faite par écrit ? Les opinions divergent. Nous rejoignons les commentateurs qui sont d'avis que la renonciation n'est pas soumise à la règle de l'écrit probatoire. Cette exigence ne s'applique qu'aux conventions et non aux actes unilatéraux tels que les renonciations²⁸².

La doctrine majoritaire estime également qu'une renonciation peut être tacite pour autant qu'elle soit certaine²⁸³. Par exemple, un contrat d'adaptation d'une œuvre soumis au régime général du droit d'auteur implique nécessairement une renonciation partielle de l'auteur à son droit à l'intégrité. En raison de l'objet du contrat, l'auteur accepte tacitement que certaines modifications soient apportées à son œuvre²⁸⁴.

Bien que la règle de l'écrit probatoire ne s'applique pas, les parties à un contrat de cession de droits patrimoniaux devront veiller à organiser contractuellement l'exercice des droits moraux. Certaines clauses parfaitement licites sont envisageables pour autant qu'elles n'entraînent pas une renonciation globale à l'exercice futur du droit assimilable à une aliénation des droits moraux. Ainsi, un auteur pourra accepter que son nom ne soit pas apposé sur son œuvre pour certaines exploitations ou accepter des altérations dues à des contraintes techniques.

Ces clauses impliquant une renonciation seront toujours interprétées de manière stricte. En cas de doute sur la portée d'une renonciation prévue dans un contrat, la clause sera interprétée strictement en faveur de l'auteur (article 3 § 1 al. 3 LDA).

Enfin, rappelons que le droit à l'intégrité contient un « noyau dur » que l'auteur conserve même en cas de renonciation tacite ou expresse. Il s'agit du droit pour l'auteur de s'opposer à toute atteinte qui serait préjudiciable à son honneur ou à sa réputation (art. 1^{er}, § 2, al. 7, LDA).

²⁸¹ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 91, n° 161.

²⁸² En ce sens : A. STROWEL, E. DERCLAYE, *op. cit.*, p. 105, n° 114 ; A. BERENBOOM, *op. cit.*, p. 146, n° 103. *Contra* : V. CASTILLE, « Les droits moraux et la bande dessinée », in *Le droit d'auteur et la bande dessinée*, Bruylant, 1997, p. 294 ; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 148, n° 83 ; A. STROWEL, « Le régime des œuvres audiovisuelles dans la loi relative au droit d'auteur », *Ing. Cons.*, 1995, p. 328.

²⁸³ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 105, n° 114 et p. 90, n° 61 ; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 148, n° 183. *Contra* : A. BERENBOOM, *op. cit.*, p. 146, n° 103.

²⁸⁴ V. CASTILLE, *op. cit.*, p. 296 ; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 158, n° 201.

B. La renonciation au droit moral et les logiciels

176. Le programmeur se voit reconnaître des droits moraux par la loi via un simple renvoi à la Convention de Berne. La loi ne définit pas les caractéristiques des droits moraux du programmeur.

Les droits moraux du programmeur sont-ils inaliénables ? Peut-il y renoncer ? Ces questions divisent la doctrine.

Certains plaident en faveur d'une interprétation du régime des droits moraux à la lumière de la Convention de Berne à laquelle la loi spéciale renvoie²⁸⁵. Toutefois, cette convention ne se prononce pas sur le caractère inaliénable ou non des droits moraux. L'interprétation de la Convention à cet égard diverge. Ainsi, selon certains, le caractère aliénable pourrait être déduit du rejet d'une proposition française voulant déclarer l'inaliénabilité des droits moraux.

Par contre, une partie de la doctrine admet l'extension du régime général de la loi sur le droit d'auteur au régime de la loi spéciale²⁸⁶. Par conséquent, les droits moraux des programmeurs disposent des mêmes caractéristiques que celles qui s'attachent aux prérogatives morales des autres créateurs. Une renonciation partielle est dès lors licite dans les limites prévues par la loi générale. Seraient licites les clauses par lesquelles le programmeur renonce globalement à exercer un droit moral à l'égard d'atteintes déjà portées ou les clauses par lesquelles il renonce, même pour l'avenir, à des atteintes préalablement déterminées ou déterminables. Par exemple, le programmeur renonce à décider seul quand le programme est achevé et divulguable à partir du moment où le programme remplit certaines fonctionnalités convenues.

Selon nous, le législateur belge n'a pas souhaité déroger au principe de l'inaliénabilité des droits moraux à l'égard des prérogatives des programmeurs. L'argument de l'aliénabilité du droit moral semble uniquement basé sur une interprétation contestée de la Convention de Berne. En cas de doute sur la volonté du législateur, le régime de la loi générale qui constitue le droit commun en la matière doit, à notre avis, l'emporter. L'assimilation des programmes d'ordinateurs aux œuvres littéraires, prévu par la loi belge va d'ailleurs en ce sens. En tout état de cause, le régime du droit moral de la loi générale est

²⁸⁵ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 212, n° 213 ; A. STROWEL, « La loi belge du 30 juin 1994 sur les programmes d'ordinateur : vers un droit d'auteur *sui generis* ? » , *R. I. D. A.*, n° 164, 1995, p. 201 ; C. DOUTRELEPONT, in *Doc. Parl.*, Chambre, s.e., 1992-1993, 1071/6, p. 12.

²⁸⁶ A. PUTTEMANS, *op. cit.*, p. 775 ; F. BRISON et J.-P. TRIAILLE, (1995), p. 143 ; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 229, n° 269.

suffisamment souple et équilibré pour être transposé aux programmes d'ordinateur.

C. La renonciation au droit moral et les logiciels libres

177. Selon le modèle du logiciel libre, le développeur cède sous licence l'exercice de l'ensemble de ses droits patrimoniaux. Le développeur conserve néanmoins ses droits moraux.

Les droits moraux peuvent constituer, dès lors, un obstacle pour une pleine utilisation du logiciel distribué sous licence libre. Le développeur pense pouvoir librement modifier le logiciel ainsi distribué alors qu'il reste tenu de respecter les droits moraux des précédents programmeurs.

La cession des droits moraux n'étant pas envisageable, la seule solution juridique est de faire renoncer préalablement l'auteur à l'exercice de ses droits moraux. Cette renonciation peut être insérée dans les licences des logiciels libres.

Deux limites imposées par la loi sont à respecter : la renonciation ne peut être globale et porter sur l'exercice futur du droit. Seules les renonciations qui présentent cumulativement ces deux caractéristiques sont nulles²⁸⁷.

Des aménagements contractuels aux droits moraux sont envisageables. Dans le cadre du développement d'un logiciel par une communauté de programmeurs, le droit de divulgation peut ainsi être modalisé contractuellement ou organisé selon une procédure précise (*supra*, n° 165). Le développeur d'un logiciel renonce partiellement à choisir le moment et la manière dont le logiciel libre sera diffusé.

§ 2. L'abus de droit moral

A. Principe

178. Un droit moral est susceptible d'être exercé abusivement. L'abus de droit est actuellement considéré comme un principe général de droit qui permet de sanctionner l'usage abusif d'un droit²⁸⁸.

²⁸⁷ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 89, n°161; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 147, n° 183.

²⁸⁸ C. CARON, *Abus de droit et droit d'auteur*, Collection le droit des affaires, Paris, Litec, 1998.

La théorie de l'abus de droit se base sur une mise en balance des intérêts en présence dans l'exercice de toute liberté ou de tout droit subjectif. Dans son arrêt de principe du 10 septembre 1971, la Cour de cassation retient le critère suivant en matière contractuelle et extra-contractuelle : « ... *l'abus de droit peut résulter de l'exercice d'un droit qui dépasse manifestement les limites de l'exercice normal de ce droit par une personne normalement prudente et diligente* »²⁸⁹.

La doctrine et la jurisprudence ont développé une série de critères particuliers habituellement reconnus et appliqués comme critères de l'abus de droit. Ainsi, abuse de son droit celui qui l'exerce dans l'intention de nuire ou qui l'exerce de la façon la plus dommageable sans avantage particulier, ou lorsque cet exercice cause un préjudice disproportionné par rapport à l'avantage recherché, ou encore lorsque le droit est exercé sans intérêt légitime ou en le détournant de sa finalité²⁹⁰.

En principe, le titulaire d'un droit moral a un intérêt légitime à exercer son droit moral. Cet exercice n'est toutefois pas discrétionnaire et ne peut conduire à un détournement ou un exercice abusif. Il incombe au défendeur de démontrer cet exercice abusif.

Ainsi, un auteur abuse de son droit moral s'il le détourne de sa finalité en vue notamment de s'assurer une rémunération ou des avantages disproportionnés²⁹¹.

B. L'abus de droit moral et les logiciels

179. La question de l'exercice abusif du droit moral dans l'univers informatique est rarement évoquée.

A notre connaissance, différentes explications peuvent être apportées. Dans une logique du logiciel propriétaire, ce sont principalement les droits patrimoniaux qui sont invoqués par le développeur pour des copies du logiciel. On n'a recours aux droits moraux qu'à titre subsidiaire. Cet état de fait est logique dans la mesure où les logiciels sont davantage des œuvres fonctionnelles et que les droits moraux reconnus au programmeur sont limités.

²⁸⁹ Cass, 10 septembre 1971, *R.C.J.B.*, 1976, p. 324, note de P. VAN OMMESSLAGHE ; J.-F. ROMAIN, *Théorie critique du principe général de bonne foi en droit privé. Des atteintes à la bonne foi, en général, et de la fraude, en particulier (fraus omnia corrumpit)*, Bruylant, Bruxelles, 2000, p. 896.

²⁹⁰ J.-F. ROMAIN, *Ibid.*, p. 900 ; C. CARON, « France : Les limites externes au droit d'auteur (abus de droit et droits du public) », Journées d'études de l'ALAI 1998, Cambridge, p. 238.

²⁹¹ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 149, n° 185 ; A. BERENBOOM, *op. cit.*, p. 139 ; Cass, 10 septembre 1971, *R.C.J.B.*, 1976, p. 324, note de P. VAN OMMESSLAGHE ; Prés. Civ. Bruxelles (cess.), 21 février 1997, *A.&M.*, 1997, p. 288.

Toutefois, on pourrait envisager le cas où l'interdiction découlant de l'exercice de son droit moral par l'auteur aurait des conséquences pratiques disproportionnées. Ainsi, selon nous, pourrait être considérée comme abusive la demande d'un auteur de retirer d'un marché déterminé tous les supports du programme sur lesquels son nom n'aurait pas été apposé. Il pourrait néanmoins exiger que son nom figure à l'avenir sur les nouveaux supports du logiciel.

Rappelons que l'existence d'un contrat de droit d'auteur, même s'il ne contient pas de clause de renonciation, influence la possibilité d'exercice des droits moraux. Après avoir conclu un contrat, le développeur ne pourra exercer ses droits moraux de manière discrétionnaire. En effet, en vertu de l'article 1134, alinéa 3, du Code civil, les parties à un contrat sont tenues d'exécuter de bonne foi la convention. Ce principe interdit aux parties d'abuser des prérogatives qu'elles puisent dans le contrat. Toutefois, un titulaire d'un droit peut exercer celui-ci dans son intérêt personnel pour autant que cet exercice ne cause pas à un tiers des avantages disproportionnés²⁹².

C. L'abus de droit moral et les logiciels libres

180. Selon l'économie du logiciel libre, le développeur se dépossède de l'ensemble de ses droits patrimoniaux. En d'autres termes, il permet que son logiciel soit complètement modifié et rend ainsi plus importante la probabilité d'atteinte à ses droits moraux. Ces derniers droits constituent la dernière base légale permettant à l'auteur d'agir à l'encontre du licencié en dehors de l'hypothèse du non respect des termes de la licence.

La tentation sera grande pour lui de se baser sur ses prérogatives morales afin de revendiquer une rémunération ou d'autres avantages commerciaux disproportionnés. La théorie de l'abus de droit constituera un garde-fou précieux pour éviter toute exagération de l'auteur dans l'exercice de ses droits.

L'appréciation de l'existence d'un abus de droit devra se faire en tenant compte de la licence du logiciel libre. L'existence d'une licence restreint de manière significative les droits moraux du programmeur. En effet, le modèle des licences de logiciels libres est d'autoriser une large possibilité de modification, de reproduction ou de distribution. Selon nous, l'exercice des droits moraux en est corollairement réduit. Les cours et tribunaux pourraient dès lors accueillir plus favorablement le moyen tiré de l'abus de droit à l'encontre d'un développeur d'un logiciel libre qui agirait sur la base de ses prérogatives morales.

²⁹² J-L. FAGNART, « L'exécution de bonne foi des conventions : un principe en expansion », *R.C.J.B.*, 1986, p. 316.

CHAPITRE 2 : LE LOGICIEL LIBRE ET LES RÈGLES CONTRACTUELLES GÉNÉRALES EN MATIÈRE DE DROIT D'AUTEUR

Section 1 : Considérations générales

181. La LDA établit un formalisme et une règle d'interprétation en matière de cession de droit d'auteur (art. 3, LDA). En l'absence de règles spécifiques dans la LPO, les conditions de formes exigées par la loi générale sont également applicables dans le domaine des logiciels.

Deux catégories de règles existent en matière contractuelle en droit d'auteur : des règles générales, qui s'appliquent à tous les contrats (art. 3, LDA), et des règles particulières à l'égard de certaines catégories de contrats tels que notamment les contrats d'emploi et les contrats de commande (art. 3, § 3, LDA)²⁹³.

Ces règles contractuelles se veulent protectrices des intérêts de l'auteur considéré habituellement comme la partie faible au contrat. Ces règles sont impératives et entraînent, en principe, la nullité des dispositions ou des conventions qui y seraient contraires²⁹⁴.

Rappelons que le terme « cession » dans la loi belge couvre les différentes formes d'exploitation des droits d'auteur. Ainsi, une cession peut couvrir à la fois une aliénation d'un droit patrimonial ou l'octroi d'une licence de ce même droit (art. 3, § 1^{er}, al. 1, *in fine*).

En outre, ces règles protectrices bénéficient en principe uniquement à l'auteur, en tant que personne physique, qui est le titulaire originaire des droits. En d'autres termes, les titulaires dérivés (cessionnaire, licencié,...) ne peuvent les invoquer à leur avantage.

Cette limite est essentielle en matière de licences de logiciels libres. En effet, le développeur d'un logiciel libre ne pourra exiger le bénéfice de ces règles protectrices que dans le cadre de l'apport au programme dont il est effectivement l'auteur. Les dispositions légales analysées dans ce chapitre ne

²⁹³ Des dispositions spécifiques sont également prévues à l'égard du contrat d'édition (articles 25 à 30 LDA), du contrat de représentation (articles 31 à 32 LDA), et du contrat d'adaptation audiovisuelle (article 17 LDA) ou du contrat de production audiovisuelle (articles 18 à 19 LDA).

²⁹⁴ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 100, n° 109.

s'appliqueront pas si une licence libre est conclue par un cessionnaire de droits qui n'est pas l'auteur de tout ou partie du programme.

Dans le système de licences de logiciel libre, il est fréquent qu'un programmeur endosse le double rôle de licencié et de donneur de licence. En effet, après avoir modifié le logiciel donné en licence, le développeur devient à son tour auteur et, éventuellement, donneur de licence. Ce n'est qu'en tant qu'auteur qu'il bénéficiera du régime protecteur de la loi.

Par ailleurs, soulignons que le droit commun et, en particulier, les règles générales relatives à la formation, à l'interprétation et l'exécution des contrats sont d'application. Outre le principe général de l'exécution de bonne foi des conventions (art. 1134, al. 3, C. civ.), il convient de rappeler que les contrats obligent « *à toutes les suites que l'équité, l'usage ou la loi donnent à l'obligation d'après sa nature* » (art. 1135, C. civ.).

Section 2 : L'écrit à titre probatoire

§ 1. Principe (art. 3. § 1^{er}, al. 2, LDA)

182. La loi dispose qu'« *à l'égard de l'auteur, tous les contrats se prouvent par écrit* » (art., 3 § 1^{er}, al. 2, LDA). Cette condition de l'écrit ne concerne que la preuve de la convention et ne conditionne pas la validité du contrat²⁹⁵. En d'autres termes, le cocontractant d'un auteur doit apporter la preuve par écrit que ce dernier lui a cédé ou consenti en licence ses droits patrimoniaux ainsi que les conditions fixées par le contrat.

L'écrit ne doit pas nécessairement prendre la forme d'un document contractuel (*instrumentum*). D'autres formes peuvent être admises telles qu'un échange de lettres ou de courriels, des conditions générales ou un bon de commande²⁹⁶. Le droit commun des obligations devra néanmoins être respecté pour que l'engagement de l'auteur de céder ses droits soit valable.

Pour mémoire, cette règle de l'écrit ne bénéficie en principe qu'à l'auteur entendu comme la personne physique qui a créé l'œuvre et non le titulaire dérivé des droits. En revanche, l'auteur pourra établir l'existence et le contenu du contrat à l'égard des tiers par toutes voies de droit²⁹⁷.

²⁹⁵ H. VANHEES, « De nieuwe wettelijke regeling inzake auteurscontracten », *R.D.C.B.*, 1995, p. 735 ; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 313, n° 393.

²⁹⁶ A. BERENBOOM, *op. cit.*, n° 120 et la jurisprudence citée.

²⁹⁷ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 313, n° 393.

La règle de l'écrit probatoire semble écarter toute possibilité d'engagement tacite de la part de l'auteur originaire. Toutefois, un certain assouplissement est admis par les cours et tribunaux avec l'application de la théorie de l'apparence²⁹⁸.

La théorie de l'apparence a été définie comme « *celle en vertu de laquelle la personne qui traite avec une autre sur la base d'une situation apparente, qui ne correspond pas à la situation réelle mais en croyant légitimement que tel est cependant le cas, va pouvoir se prévaloir des effets qu'aurait créé la situation apparente si elle avait été réelle* »²⁹⁹.

Différentes conditions doivent être réunies pour l'application de la théorie de l'apparence³⁰⁰.

La première condition est l'existence d'une situation apparente différente de la situation réelle. La deuxième condition requiert que la création de la situation apparente soit imputable à celui à qui on veut en faire assumer les effets. La troisième est que la victime doit être de bonne foi. En d'autres termes, la victime de l'apparence doit avoir la croyance légitime que la situation apparente correspond à la situation réelle. Une dernière condition, plus controversée, est celle de savoir si l'application de la théorie de l'apparence requiert que le tiers victime subisse un dommage si la situation ostensible ne se voit pas reconnaître un effet réel³⁰¹.

Consacrée par la Cour de cassation, la théorie de l'apparence semble néanmoins difficilement conciliable avec les dispositions du droit d'auteur. Ainsi, l'exigence de l'écrit s'oppose *a priori* à ce qu'une situation apparente puisse produire des effets tels qu'une cession ou une licence de droits.

Cependant, la théorie de l'apparence est une théorie d'application générale qui impose aux titulaires de droits d'auteur d'éviter tout comportement de nature à

²⁹⁸ Cass., 20 juin 1988, *R.C.J.B.*, 1991, note R. KRUTHOF, « La théorie de l'apparence dans une nouvelle phase », p. 51 et s. ; Bruxelles (9^e ch.), 15 février 1996, *A&M.*, 1996, note de M. BUYDENS, « Réflexions sur l'application de la théorie de l'apparence en droit d'auteur », p. 407.

²⁹⁹ P. VAN OMMESLAGHE, « L'apparence comme source autonome d'obligations et le droit belge », *R.D.C.D.I.*, 1983, p. 144 ; C. VERBRUGGEN, « La théorie de l'apparence : quelques acquis et beaucoup d'incertitudes », *Mélanges offerts à Pierre Van Ommeslaghe*, Bruxelles, Bruylant, 2000, p. 301.

³⁰⁰ C. VERBRUGGEN, *op. cit.*, p. 315.

³⁰¹ P.A. FORIERS, « L'apparence, source autonome d'obligations, ou application du principe général de l'exécution de bonne foi », *J.T.*, 1989, p. 541.

créer une situation susceptible de tromper les tiers et à leur causer préjudice³⁰². La création d'apparence peut être fautive si l'auteur n'accepte pas les conséquences de la situation apparente qu'il a contribué à créer en vue de surprendre les tiers³⁰³. Comme le souligne M. BUYDENS, « *le droit des auteurs de ne pas reconnaître les cessions qui ne sont pas constatées par écrit doit céder le pas devant le principe de bonne foi qui s'impose à tous* »³⁰⁴.

L'ensemble des circonstances devront être prises en compte pour déterminer si les conditions de la théorie de l'apparence peuvent être rencontrées (expérience des parties en matière de cession de droits d'auteur, secteur d'activité habituel des parties, légitimité de la croyance de la prétendue victime, durée des relations contractuelles, etc.). L'application de cette théorie de l'apparence sera avant tout une question de fait.

§ 2. *L'écrit à titre probatoire et le logiciel libre*

183. L'exigence de la preuve écrite vaut pour les licences de logiciels libres qui sont des contrats par lesquels un donneur de licence concède des droits patrimoniaux à un licencié. Cette règle n'est d'application que si le donneur de licence est lui-même l'auteur originaire de tout ou partie du logiciel donné sous licence.

Celui ou celle qui prétend bénéficier d'une licence de type « libre », sur un logiciel doit démontrer que son auteur lui a effectivement distribué le logiciel sous ce type de licence. De manière identique, le cocontractant du développeur d'un logiciel libre devra démontrer par écrit l'existence de toute condition particulière liée à l'octroi de la licence. Ainsi, si un logiciel libre est distribué à la fois sous une version gratuite et une version payante, le licencié qui prétend détenir une licence à titre gratuit devra en apporter la preuve par écrit.

Toutes les licences de logiciels libres sont écrites. Certaines licences telles que la B.S.D. sont particulièrement succinctes et sont susceptibles de laisser des zones d'ombre. En cas de litige portant sur des domaines non réglés par la licence, le licencié risque d'éprouver des difficultés à démontrer par écrit les termes de la licence. Le bénéficiaire d'une licence sur un logiciel libre aura

³⁰² Cass., 20 juin 1988, *R.C.J.B.*, 1991, note R. KRUTHOF, « La théorie de l'apparence dans une nouvelle phase », p. 51 et s. ; VAN OMESLAGHE, *op. cit.*, p. 144 ; M. BUYDENS, « Réflexions sur l'application de la théorie de l'apparence en droit d'auteur », note sous Bruxelles (9^e ch.), 15 février 1996, *A&M.*, 1996, p. 407.

³⁰³ P.A. FORIERS, « L'apparence, source autonome d'obligations, ou application du principe général de l'exécution de bonne foi », *J.T.*, 1989, p. 543.

³⁰⁴ Bruxelles (9^e ch.), 15 février 1996, *A&M.*, 1996, note de M. BUYDENS, « Réflexions sur l'application de la théorie de l'apparence en droit d'auteur », p. 409.

intérêt à prévoir les aménagements contractuels nécessaires pour s'assurer que toutes les formalités légales sont respectées.

Pour combler les éventuelles lacunes des contrats, la théorie de l'apparence pourrait s'appliquer en considérant que l'attitude du donneur de licence crée une situation de fait qui produit des effets juridiques à son encontre. Selon nous, cette théorie pourrait s'appliquer pour certaines modalités pratiques de la licence. Toutefois, la théorie de l'apparence ne peut conduire à ignorer toutes les imprécisions d'une licence au risque d'entrer en contradiction avec le principe de la preuve par écrit à l'égard de l'auteur et le formalisme requis en matière de cession. Une application systématique de la théorie de l'apparence conduirait à notre avis à vider la loi de son sens. Un contrat sera affecté de nullité en tout ou en partie lorsque les lacunes touchent à une partie ou à l'ensemble des modes d'exploitation.

La théorie de l'apparence devra être appliquée en tenant compte de la légitime croyance du bénéficiaire de la licence en l'existence de droits consentis par le donneur de licence.

Section 3 : L'interprétation stricte

§ 1. Principe (art. 3, § 1^{er}, al. 3, LDA)

184. La loi prévoit que les dispositions contractuelles relatives au droit d'auteur et à ses modes d'exploitation sont de stricte interprétation, en faveur de l'auteur originaire.

Selon la jurisprudence, cette règle n'est d'application que si la convention est peu claire ou est sujette à des interprétations divergentes³⁰⁵. Seule la disposition contractuelle ambiguë, ou qui fait naître un doute quant à son contenu véritable, exige une interprétation stricte.

En vertu de l'article 1315 du Code civil, le cessionnaire ou le licencié doit prouver l'étendue des droits qu'il a acquis ou reçus sous forme de licence. En cas de doute sur la portée d'une cession, la règle de l'interprétation stricte privilégiera l'auteur.

³⁰⁵ Cass., 11 mai 1998, *R.W.*, 1998-1999, p. 713 et note H. VANHEES ; Cass., 13 février 1941, *Pas.*, 1942, p. 44 ; Cass., 11 novembre 1943, *Pas.*, 1944, I, p. 49 ; H. VANHEES, « De nieuwe wettelijke regeling inzake auteurscontracten », *R.D.C.B.*, 1995, p. 739, n° 28 ; B. DAUWE, « Overeenkomsten in het oude en in het nieuwe auteursrecht », *Belgisch auteursrecht van oud naar nieuw – Le renouveau du droit d'auteur en Belgique*, Bruylant, 1996, p. 260.

§ 2. *L'interprétation stricte et le logiciel libre*

185. La règle de l'interprétation stricte est d'application en cas de licence d'un logiciel libre en cas de doute sur la portée des termes. Dans ce cas, la licence d'un logiciel libre s'interprétera strictement et en faveur de l'auteur.

Cette règle peut s'avérer utile dans le cadre de logiciels libres car de nombreuses licences contiennent des clauses lacunaires ou des formulations hasardeuses.

Toutefois, le principe de l'interprétation stricte n'est d'aucune utilité à l'égard d'une disposition qui est formulée de manière générale ou absolue³⁰⁶. Ainsi, aucune interprétation n'est nécessaire à l'égard de l'article 1^{er} de la G.P.L. qui prévoit : « *Vous pouvez copier et distribuer des copies verbatim des codes sources du programme comme vous l'avez reçu, selon tous les moyens* ³⁰⁷(...) ». Une telle disposition est néanmoins susceptible d'entrer en contradiction avec les règles d'imposition de mentions légales obligatoires (Section 5).

Section 4 : L'obligation d'exploiter selon les usages honnêtes

§ 1. *Principe (art. 3, § 1^{er}, al. 5, LDA)*

186. La loi belge prévoit que le cessionnaire ou le licencié est tenu d'assurer l'exploitation de l'œuvre conformément aux usages honnêtes de la profession.

L'obligation d'exploiter variera selon l'objet du contrat et le domaine concerné. Selon la doctrine, la notion des « usages honnêtes » fait référence implicitement à la loi du 14 juillet 1991 sur les pratiques de commerce et l'information et la protection du consommateur³⁰⁸. Cette règle impose de prendre en considération à la fois les intérêts de l'auteur mais également les usages du secteur concerné. Ainsi, le refus d'exploitation de certaines parties d'une œuvre peut être justifié sur base des pratiques habituelles d'un secteur particulier (cinéma, publicité, etc.).

Cette règle est impérative et ne peut dès lors être écartée par contrat. Une clause dérogeant à l'obligation d'exploitation pourra être sanctionnée de nullité. L'auteur dispose des moyens de droit commun pour agir à l'encontre de celui

³⁰⁶ H. VANHEES, « De nieuwe wettelijke regeling inzake auteurscontracten », *R.D.C.B.*, 1995, p. 739 et s.

³⁰⁷ Traduction libre.

³⁰⁸ *M.B.* 29 août 1991, p. 5. ; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 114, n° 124 ; F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 332, n° 414.

qui demeurerait en défaut d'exploiter son œuvre. L'auteur pourrait solliciter l'exécution en nature de son obligation ou demander la résolution du contrat et des éventuels dommages et intérêts³⁰⁹.

§ 2. *L'obligation d'exploiter et le logiciel libre*

187. En matière informatique, l'obligation d'exploiter ne joue pas un rôle prépondérant. En raison de leur but fonctionnel, rares seront les logiciels qui ne seront pas exploités conformément aux usages honnêtes. Toutefois, à l'encontre d'un éditeur de logiciel qui demeurerait en défaut de publier et diffuser un logiciel, un auteur pourrait solliciter l'exécution en nature de l'obligation pesant sur lui ou, à défaut, la résolution du contrat à ses torts³¹⁰.

En ce qui concerne le domaine du logiciel libre, l'exploitation conformément aux usages honnêtes n'est pas aisée à définir. En effet, la variété des usages et des types de licences ne permet pas d'établir des pratiques acceptées dans ce secteur.

Un donneur de licence d'un logiciel libre souhaite une exploitation du logiciel sans toutefois l'exiger comme une contrepartie de la cession de ses droits sur son œuvre. Cette exploitation comportera parfois l'obligation de conserver le caractère libre de la licence. Dans cette hypothèse, il incombera à l'utilisateur de veiller à conserver ce caractère libre par tous les moyens sous peine non seulement de ne pas tenir compte des termes de la licence mais également de ne pas respecter son obligation d'exploiter selon les usages honnêtes.

Par ailleurs, la non exploitation d'un logiciel sous licence libre pourrait être considérée comme un usage honnête. Les pratiques du secteur du logiciel libre sont étrangères à toute idée d'obligation d'exploiter le logiciel donné sous licence.

Section 5 : Les mentions obligatoires

§ 1. *Principe (art. 3, § 1^{er}, al. 4, LDA)*

³⁰⁹ Pour mémoire, cette règle ne bénéficie qu'à l'auteur originaire et non aux titulaires dérivés. Aucune obligation d'exploitation ne s'impose dans les contrats conclus ultérieurement où l'auteur n'est pas partie.

³¹⁰ A. BERENBOOM, *op. cit.*, n° 129; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 114, n° 124.

188. La loi prévoit que « *pour chaque mode d'exploitation, la rémunération de l'auteur, l'étendue et la durée de la cession doivent être déterminées expressément* ». Cette règle de forme ne bénéficie en principe qu'à l'auteur et non au cessionnaire ou au licencié.

Ce formalisme permet à l'auteur de mesurer l'étendue des droits qu'il cède et de lui assurer une rémunération pour chaque mode d'exploitation. Cette clause tend à garantir l'information de l'auteur et la possibilité de négocier avec le cocontractant les conditions de la cession de manière transparente.

La loi détermine les éléments qui doivent expressément figurer dans les contrats d'exploitation. Ainsi, les mentions exigées par la loi sont au nombre de quatre : la mention du mode d'exploitation visé, la rémunération de l'auteur, l'étendue et la durée de la cession.

A. Modes d'exploitation

189. La notion de « mode d'exploitation » n'est pas définie par la loi. Le niveau de précision requis dans la mention du mode d'exploitation est controversé. Les travaux préparatoires de la loi précisent qu'« *il n'est cependant pas nécessaire que chaque mode distinct d'exploitation soit spécifié de manière détaillée* »³¹¹.

La doctrine admet généralement que chaque droit patrimonial, tel que prévu par la loi, doit être mentionné (droit de reproduction, droit de communication au public, droits de traduction et d'adaptation, droits de location et/ou de prêt)³¹². En outre, un degré de précision supplémentaire est généralement requis quant au type de support (livre, cd-rom, disque, etc.) ou quant à la nature de l'exploitation (adaptation graphique ou musicale, audiovisuelle, etc.). En réalité, le niveau de précision dans la description du mode d'exploitation dépendra du secteur, du type d'œuvre et des usages professionnels concernés³¹³.

B. Rémunération

³¹¹ Selon les travaux préparatoires, la notion de forme d'exploitation renvoie à celle d'une technique alors que celle de « mode d'exploitation » relève de la notion d'acte d'exploitation au sens large. Rapport DE CLERCK, *op. cit.*, p. 120. Voy. également : F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 321, n° 401.

³¹² F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 321, n° 401; A. BERENBOOM, *op. cit.*, n° 122; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 113.

³¹³ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 323, n° 401, *in fine*; A. BERENBOOM, *op. cit.*, n° 122, *in fine*; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 109, n° 119.

190. La rémunération peut être forfaitaire ou fixée proportionnellement selon chaque mode d'exploitation³¹⁴. Il peut s'agir d'un montant unique et global pour l'ensemble des modes d'exploitation pour autant que cela soit clairement indiqué dans la convention³¹⁵. Enfin, rien n'empêche les parties de fixer une rémunération nulle dans le contrat³¹⁶.

C. Etendue géographique

191. L'étendue vise les territoires géographiques sur lesquels la licence est concédée. Ainsi, les parties peuvent valablement convenir que l'étendue géographique vise « tous pays » ou « le monde entier »³¹⁷.

D. Durée

192. La durée peut être fixée librement par les parties à condition qu'elle soit dûment précisée dans le contrat. Cette durée conventionnelle peut concerner indistinctement l'ensemble des modes d'exploitation³¹⁸. Les parties peuvent valablement convenir d'une durée illimitée ou égale à la durée de la propriété intellectuelle (en principe 70 ans après le décès de l'auteur).

E. Sanction en cas de défaut d'une mention

193. La loi ne prévoit expressément aucune sanction en cas de défaut d'une des mentions dans le contrat³¹⁹.

En principe, la clause sera nulle dans la mesure où elle ne respecte pas une règle impérative prévue par le législateur en faveur de l'auteur. Si une des mentions est absente pour un mode d'exploitation déterminé, la cession pour ce mode déterminé sera réputée ne pas avoir eu lieu. En pratique, les différentes mentions exigées deviennent des règles de validité³²⁰.

³¹⁴ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 323, n° 402; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 111, A. BERENBOOM, *op. cit.*, n° 123.

³¹⁵ A. BERENBOOM, *op. cit.*, n° 123.

³¹⁶ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 111.

³¹⁷ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 324, n° 403; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 111; A. BERENBOOM, *op. cit.*, n° 124.

³¹⁸ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 324, n° 404 ; A. STROWEL, E. DERCLAYE, *op. cit.*, p. 111; A. BERENBOOM, *op. cit.*, n° 124.

³¹⁹ La mention manquante peut être déduite des autres mentions du contrat.

³²⁰ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 324, n° 400.

La nullité portera sur l'ensemble du contrat si le mode d'exploitation concerné est un élément essentiel ou déterminant du contrat. En revanche, la nullité sera partielle si le non-respect de la règle des mentions obligatoires est circonscrit à un mode d'exploitation déterminé sans porter atteinte à l'essence même du contrat³²¹.

Une partie de la doctrine adopte une position plus nuancée en distinguant le défaut d'un mode d'exploitation et le défaut d'une mention relative à un mode d'exploitation (durée, étendue et rémunération de l'auteur)³²². La sanction applicable est susceptible de varier selon le type de lacune.

Un mode d'exploitation non mentionné sera écarté du champ de la cession en vertu de l'application de l'interprétation stricte. En revanche, l'absence d'une mention susmentionnée relative à un mode d'exploitation ne pourra conduire à écarter un mode d'exploitation qu'en cas de doute sur la volonté des parties d'avoir voulu effectivement écarter ce mode d'exploitation. Les autres clauses du contrat et tout autre document entourant le contrat aideront à déterminer la volonté des parties. Selon nous, cette application modulée des sanctions semble davantage respectueuse de la volonté des parties, tout en respectant le prescrit de la loi.

§ 2. *Les mentions obligatoires et le logiciel libre*

194. Examinons comment les exigences de forme requises par la loi sont prises en compte dans les licences de logiciels libres.

A. Modes d'exploitation

195. La plupart des licences de logiciels libres indiquent, avec plus ou moins de précision, les droits patrimoniaux que l'auteur entend donner en licence.

La G.P.L. autorise la copie, la modification (incluant la traduction) ainsi que la distribution du logiciel libre (articles 0, 1 et 2 G.P.L.) mais exclut tout autre mode d'exploitation (article 0, al. 2).

La B.S.D. permet également la modification, la distribution et l'utilisation du logiciel.

³²¹ A. STROWEL, B. STROWEL, « La nouvelle législation sur le droit d'auteur », *J.T.*, 1995, p. 128, n° 46, *in fine*; B. DAUWE, *op. cit.*, p. 260-261.

³²² A. STROWEL, E. DERCLAYE, *op. cit.*, p. 112-113, n° 122.

Les licences les plus complètes sont l'O.S.L. et la Q.P.L. La première autorise explicitement la reproduction, la réalisation d'œuvres dérivées, la distribution de l'œuvre originale, l'exécution et la communication de l'œuvre au public (article 1 O.S.L.). La seconde consacre une section, intitulée « granted rights », qui décrit l'ensemble des droits consentis par le donneur de licence de manière détaillée (articles 1 à 6 Q.P.L.). Ces licences octroient des droits souvent de manière générale. On pourrait considérer que le degré de précision requis n'est pas toujours atteint. Certaines licences omettent notamment de mentionner les supports, la nature du mode d'exploitation cédé ou le type d'utilisation autorisée.

Rappelons néanmoins que la description du mode d'exploitation doit être appréciée selon le secteur, le type d'œuvre et les usages de la profession concernée.

En vertu de l'article 1156 du Code civil, il convient toujours de se référer à l'intention commune des parties pour apprécier l'étendue de la cession de droits. En cas de doute sur l'intention des parties, le principe de l'interprétation stricte jouera.

Le programmeur qui distribue son logiciel sous une licence libre vise à permettre une exploitation étendue du logiciel par le bénéficiaire de la licence. La formulation large des clauses est justifiée par la volonté du programmeur originaire qui n'entend pas s'embarrasser d'une description de tous les modes d'exploitation. Une interprétation conforme à la volonté des parties pourrait conduire à une appréciation souple de l'obligation légale de mentionner les mode d'exploitation en matière de logiciels libres. Toutefois, admettre une cession en des termes trop généraux conduirait à autoriser ce que la LDA vise précisément à empêcher.

B. Rémunération

196. A l'exception de la licence O.S.L., aucune des licences ne contient une clause spécifiquement consacrée à la rémunération de l'auteur.

Dans son préambule, la G.P.L. rappelle néanmoins qu'en anglais, le terme « free » se rapporte à libre et non pas à « gratuit » et, par conséquent, le service de distribution du logiciel peut être payant (Préambule G.P.L.). La G.P.L. traite à l'article 11 de l'exonération de responsabilité du programmeur à l'égard d'un éventuel défaut présent du logiciel. Cette exonération est justifiée par le fait que la licence est gratuite (« free of charge ») (article 11 G.P.L.). Cette contradiction de la G.P.L. fait naître une certaine confusion sur le caractère gratuit ou non de cette licence.

Par ailleurs, sur le site web de distribution de la G.P.L., les concepteurs prévoient que la distribution des copies peut être payante. Il s'agit, à notre avis, davantage d'un paiement relatif à la vente de la copie « matérielle » d'un exemplaire du logiciel que d'une rémunération pour la cession d'éventuels droits d'auteur³²³.

L'O.S.L. prévoit que la licence est octroyée libre de toute rémunération (« royalty-free ») à condition que toute sous-licence soit également octroyée à titre gratuit (article 1 O.S.L.).

C. Etendue géographique

197. Mise à part l'O.S.L., aucune référence n'est faite quant à l'étendue géographique de l'exploitation conférée dans les licences. L'O.S.L. prévoit que les droits sont consentis pour le monde entier (« worldwide ») (article 1 O.S.L.).

D. Durée

198. Les licences de logiciels libres reprises dans notre analyse ne mentionnent aucune durée à l'exception de l'O.S.L. qui prévoit une durée perpétuelle (article 1 O.S.L.).

E. Sanction en cas de défaut d'une mention

199. La majorité des licences présente des lacunes ou des approximations par rapport au formalisme exigé par la loi en matière de cession de droit d'auteur. Seule l'*Open Software License* fait figure d'exception notable en reprenant l'ensemble des exigences légales. L'absence de mentions dans les autres licences résulte sans doute de la forte influence du droit américain lors de la rédaction des licences. En outre, ces licences ont été plus souvent rédigées par des informaticiens que par des juristes.

Quelle est la sanction de ces lacunes dans les différentes licences ?

En principe, la licence pourrait être affectée de nullité car ces lacunes formelles touchent à tous les modes d'exploitation. Cette nullité sera toutefois la solution extrême qui ne sera choisie qu'en cas d'impossibilité de pouvoir déterminer la rémunération, l'étendue géographique ou la durée sur d'autres bases. D'autres

³²³ Voyez le site des concepteurs de la G.P.L. : <http://www.gnu.org/philosophy/selling.fr.html>.

clauses ainsi que les documents entourant le contrat pourront éventuellement aider à rechercher l'intention des parties et combler les lacunes des licences. Ainsi, la durée de la licence pourra notamment être déduite de la durée du contrat dans son ensemble.

En l'absence de tout moyen pour déterminer les mentions légales, les licences de logiciels libres risquent de se voir annulées. Le risque demeure sans doute théorique car la nullité demeure relative ce qui implique que seul l'auteur puisse l'invoquer. Or, l'auteur hésitera à l'invoquer en pratique car cela conduit à l'illégalité de son logiciel. En effet, il n'était autorisé à reprendre l'œuvre antérieure que dans le respect des conditions de la licence libre. Si cette dernière est nulle, il n'a plus d'autorisation et son œuvre dérivée ne pourra plus être exploitée.

De manière générale, les licences analysées devraient à notre avis être révisées pour inclure les mentions précitées. Ainsi, les différents modes d'exploitation pourraient être mentionnés de manière plus exhaustive et les licences devraient indiquer, par défaut, que les cessions sont gratuites, perpétuelles et valables pour le monde entier. Tels semblent être la volonté et l'esprit des rédacteurs de la majorité des licences. Les parties resteraient évidemment libres de fixer d'autres dispositions contractuelles.

Section 6 : Les formes d'exploitation encore inconnues

§ 1. Principe (art. 3, § 1^{er}, al. 6, LDA)

200. La loi énonce que « *la cession des droits concernant des formes d'exploitation encore inconnues est nulle* ». La *ratio legis* de cette disposition est d'empêcher qu'un auteur puisse s'engager contractuellement à propos de formes d'exploitation de son œuvre dont il ignore l'existence au moment de la signature du contrat. Cette clause permet d'assurer une rémunération de l'auteur pour des exploitations de l'œuvre qu'il ne peut raisonnablement connaître lors de la conclusion du contrat. Il s'agit d'une appréciation objective du caractère inconnu du point de vue du public en général³²⁴.

Une forme nouvelle d'exploitation nécessitera une autorisation supplémentaire de l'auteur et permettra à ce dernier de (re)négocier les conditions de son contrat initial ou de conclure un nouveau contrat.

³²⁴ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 327, n° 406.

Toutefois, certains auteurs estiment qu'une cession relative à une forme d'exploitation inconnue est valable si elle est implicitement contenue dans un des modes d'exploitation prévu par le contrat. Ainsi, un contrat d'édition musicale qui autorise la production de disques de manière analogique permet également d'éditer des disques en mode numérique³²⁵. En revanche, toute forme d'exploitation qui n'est pas raisonnablement incluse dans un des modes d'exploitation ne pourra être cédée par l'auteur. Il convient de ne pas aller au-delà de ce que l'auteur entendait céder au moment de la conclusion du contrat. Ainsi, un contrat d'édition musicale qui prévoit la distribution de musique sous forme de disques ne comprend pas la distribution de musique par Internet.

Cette interprétation nous paraît complexe à mettre en œuvre et discutable en matière de droit d'auteur. La distinction entre une forme d'exploitation incluse ou non dans un autre mode d'exploitation relèvera souvent d'une appréciation délicate et subjective qui risque d'être défavorable à l'auteur. En outre, il convient de rappeler la règle de l'interprétation stricte qui privilégie l'intérêt de l'auteur. Or, considérer qu'une forme d'exploitation est implicitement contenue dans un mode d'exploitation constitue une interprétation en faveur du cocontractant de l'auteur et non de ce dernier. Sous peine de vider l'obligation légale de sa substance, il convient de se tenir à la règle de la nullité de toute forme d'exploitation encore inconnue.

§ 2. *Les formes d'exploitation encore inconnues et le logiciel libre*

201. Les licences de logiciels libres ne peuvent concerner que des formes d'exploitation qui sont connues au moment où le programmeur distribue son logiciel sous licence.

Les licences de logiciels libres analysées ne contiennent aucune disposition expresse quant aux formes d'exploitation encore inconnues des logiciels. La formulation large des clauses de cession de certaines licences libres pourrait être interprétée comme octroyant des droits sur des formes d'exploitation encore inconnues.

Cependant, en application de la règle précitée conjuguée avec le principe de l'interprétation stricte, seuls les modes d'exploitation mentionnés dans le contrat de licence sont à notre avis effectivement cédés.

L'article 9 de la G.P.L. permet à la *Free Software Foundation* de publier des versions nouvelles ou modifiées de la licence. En principe, chaque version de la G.P.L. est identifiée par un numéro. Les parties sont libres de choisir leur propre

³²⁵ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 326, n° 405.

version de G.P.L. Si la version de la G.P.L. n'est pas précisée, le bénéficiaire de la licence est libre de décider quelle version de la licence s'applique au programme.

Ce choix unilatéral réservé au bénéficiaire est non seulement discutable, au niveau notamment du consentement des parties, mais semble également être en contradiction avec la loi sur le droit d'auteur. En effet, le bénéficiaire de la licence est susceptible de choisir une version de la G.P.L. postérieure à celle où le logiciel a été distribué. Si cette nouvelle version de la G.P.L. comprend des formes d'exploitation encore inconnues au temps de la distribution, le donneur de licence se verra imposer une licence qui comprend une forme d'exploitation dont il ignorait l'existence au moment de la distribution de son logiciel. En application de la loi, cette cession sur une forme d'exploitation inconnue du logiciel serait nulle³²⁶.

Section 7 : La cession de droits sur des œuvres futures

§ 1. Principe (art. 3, § 1^{er}, al. 5, LDA)

202. La loi prévoit que « *la cession des droits patrimoniaux relatifs à des œuvres futures n'est valable que pour un temps limité et pour autant que les genres des œuvres sur lesquelles porte la cession soient déterminés* ».

Cette règle autorise la cession des droits sur une œuvre future pourvu qu'elle soit consentie pour une période déterminée ou déterminable et que le « genre » de l'œuvre soit également précisé.

³²⁶ A. METZGER et T. JAEGER, *op. cit.*, p. 67 et 68.

§ 2. *La cession de droits sur des œuvres futures et le logiciel libre*

203. Le bénéficiaire d'une licence de logiciel libre s'engage à céder ses droits sur les modifications qu'il apporterait à l'œuvre en cas de distribution de la version modifiée du programme.

C. CARON soutient que les licences de logiciels libres réalisent dès lors une cession illicite de droits sur une œuvre future dans le chef du licencié³²⁷. En effet, le licencié renoncerait anticipativement à exercer ses droits patrimoniaux sur l'œuvre dérivée constituée du logiciel donné sous licence et de ses modifications originales.

Le licencié ne s'engage, à notre avis, à céder ses droits sur ses modifications que dans l'hypothèse où il souhaite distribuer la version modifiée du logiciel. Le programmeur conserve ses droits pour toute autre utilisation indépendante de ses modifications. Par exemple, il pourra distribuer sous licence propriétaire et de manière séparée un fichier qui ajoute une fonctionnalité supplémentaire au logiciel (appelé en anglais « patch »)³²⁸.

La modification du logiciel peut être comparée à l'adaptation d'une œuvre littéraire. L'auteur de l'adaptation ne peut exploiter l'œuvre ainsi adaptée (« œuvre dérivée ») sans le consentement de l'auteur originaire. De la même façon, le programmeur ne pourra distribuer la version modifiée du logiciel sans le consentement du programmeur originaire. Or, ce dernier exige pour toute exploitation que le programmeur cède ses droits sous licence libre.

Selon nous, le programmeur ne cède pas ses droits concernant des œuvres futures mais doit uniquement respecter la volonté de l'auteur originaire du logiciel.

³²⁷ C. CARON, *op. cit.*, p. 1558, n° 8.

³²⁸ L'article 2 al. 2 de la G.P.L. le précise expressément : « If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it ».

Section 8 : Le contrat d'emploi

§ 1. Principe (art. 3, LPO)

204. La LPO instaure un régime dérogatoire à LDA en prévoyant que « *seul l'employeur est présumé cessionnaire des droits patrimoniaux relatifs aux programmes d'ordinateur créés par un ou plusieurs employés ou agents dans l'exercice de leurs fonctions ou d'après les instructions de leur employeur* ». Par conséquent, un employé est présumé céder ses droits patrimoniaux sur le logiciel à son employeur lorsque le logiciel est développé dans l'exercice de ses fonctions ou d'après les instructions de son employeur. Il s'agit d'une présomption simple qui ne s'exerce qu'à l'égard des droits patrimoniaux³²⁹.

Un contrat d'emploi en matière de programmes d'ordinateur ne sera pas soumis aux conditions de validité prévues pour un contrat d'emploi classique qui oblige que la cession soit expressément mentionnée dans le contrat³³⁰.

§ 2. Le contrat d'emploi et le logiciel libre

205. Dans le cadre d'un contrat de travail dont l'objet est le développement d'un programme d'ordinateur, un employé-programmeur ne pourra distribuer le logiciel qu'il a créé via une licence de logiciel libre. En effet, en application de la présomption légale de cession, l'employeur bénéficiera des droits patrimoniaux sur le logiciel réalisé par l'employé dans l'exercice de ses fonctions. Notons que le logiciel libre développé par un employé pour son employeur en dehors de ses heures de travail ou à son domicile est également susceptible d'être considéré comme conçu dans l'exercice des fonctions³³¹.

Un employeur bénéficiera également de la présomption légale si son employé participe à la réalisation d'un logiciel libre en suivant une mission qui lui a expressément été confiée. Tel est le cas du comptable qui concourt à la réalisation d'un logiciel libre de comptabilité sans être informaticien ou sans avoir un contrat de travail dont l'objet est la réalisation d'un logiciel.

Dans ces hypothèses, l'employé ne dispose plus de droits sur le logiciel développé. Par conséquent, il ne pourra ensuite céder ses droits sous licence qu'elle soit propriétaire ou libre. Rappelons que la présomption de cession ne

³²⁹ F. DE VISSCHER, B. MICHAUX, *op. cit.*, p. 320-321, n° 255-256.

³³⁰ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 207, n° 232 ; F. BRISON, J.-P. TRIAILLE, *op. cit.*, p. 141.

³³¹ A. STROWEL, E. DERCLAYE, *op. cit.*, p. 208, n° 234.

concerne que les droits patrimoniaux et non les droits moraux. En d'autres termes, l'employé qui a développé un logiciel, distribué sous licence libre ou propriétaire, conservera ses droits moraux.

En revanche, l'employeur garde la possibilité de distribuer le logiciel via la licence qu'il juge la plus appropriée. Ainsi, une société peut privilégier une licence de logiciel libre pour permettre à ses clients d'avoir accès aux codes sources des logiciels et les modifier pour les adapter à leurs besoins. Rappelons qu'en principe, un employeur ne pourra bénéficier du formalisme et des règles d'interprétation prévues en faveur de l'auteur originaire.

Section 9 : Le contrat de commande

§ 1. Principe (art. 3, § 3, al. 2 et 3, LDA)

206. En l'absence de disposition dans la LPO, le régime général de la LDA prévoit que « *les droits patrimoniaux peuvent être cédés à celui qui a passé la commande pour autant que l'activité de ce dernier relève de l'industrie non culturelle ou de la publicité, que l'œuvre soit destinée à cette activité et que la cession des droits soit expressément prévue* ».

En d'autres termes, celui qui passe commande d'une œuvre à un auteur pourra bénéficier des droits patrimoniaux sur l'œuvre à trois conditions :

- 1) La cession est expressément prévue dans le contrat
- 2) L'activité de celui qui passe la commande relève de l'industrie non culturelle ou de la publicité
- 3) L'œuvre est destinée à cette activité.

Le législateur n'a pas défini ce qu'il faut entendre par « industrie non culturelle ». Le caractère culturel ou non d'un programme d'ordinateur dépendra de l'utilisation qui en est faite. Un programme d'ordinateur est susceptible de relever de l'industrie culturelle (jeux vidéo) et relever de l'industrie non culturelle (logiciel de comptabilité). Il conviendra en général d'apprécier l'activité à laquelle est destiné le logiciel qui fait l'objet du contrat de commande. Ainsi, la commande d'un site web présentant les activités d'une maison d'édition ne suffit pas à lui conférer un caractère culturel³³².

La création réalisée en exécution d'un contrat de commande bénéficie d'un régime de formalités assoupli. Le contrat ne doit pas mentionner les modes

³³² Civ. Bruxelles (8^{ème} ch.), 23 juin 2003, A.&M., 2003, p. 385 note F. BRISON.

d'exploitation et peut concerner des œuvres futures. En outre, celui qui passe la commande ne doit pas s'engager à exploiter l'œuvre. En revanche, la cession doit faire l'objet d'un écrit et sera interprétée strictement. Une clause permettant une exploitation sous une forme inconnue est valable si elle est expressément prévue et qu'une participation au profit est stipulée.

§ 2. *Le contrat de commande et le logiciel libre*

207. Si les formes légales du contrat de commande d'un logiciel sont respectées, le programmeur peut céder ses droits ou les donner sous licence à celui qui a passé la commande. En tant que titulaire des droits, ce dernier disposera du choix de distribuer le logiciel sous forme libre ou non.

En principe, la règle de la mention expresse de la cession dans le contrat devrait suffire à attirer l'attention du programmeur sur le transfert de droits opéré par le contrat de commande.

Un programmeur qui souhaite que le logiciel soit « libre » devra négocier avec son cocontractant que la licence libre soit utilisée en cas de diffusion ou distribution du logiciel. Le contrat de commande pourra prévoir que le programme commandé soit distribué sous licence libre. Celui qui passe la commande aura le choix de conserver le logiciel pour son usage privé ou interne et, en cas de diffusion à des tiers, sera tenu d'opter pour une licence libre.

En pratique, le programmeur disposera d'une marge réduite de négociation par rapport à son cocontractant qui, en tant que commanditaire, financera le développement du logiciel et sera en mesure d'imposer son mode de distribution.

CONCLUSION

208. Le génie des licences des logiciels libres est de s'appuyer sur le droit d'auteur afin de permettre à l'utilisateur un large champ d'utilisation du logiciel. Ces licences opèrent un renversement du droit d'auteur en autorisant l'accès au code source ainsi que la libre modification et reproduction du logiciel.

En réalité, cette cession ou mise sous licence des droits ne concerne que les droits patrimoniaux des développeurs. En effet, en vertu du principe d'inaliénabilité des droits moraux, l'auteur d'un logiciel libre conserve ses droits moraux en dépit de tout contrat de cession ou de licence.

Dans les contrats analysés, un souci du respect des droits moraux du programmeur se reflète tantôt de manière explicite (droit de paternité), tantôt de manière plus diffuse (droit à l'intégrité). En revanche, le droit de divulgation, certes plus controversé en matière de logiciel, n'est pratiquement jamais pris en compte dans les licences. En règle générale, ces licences gagneraient à une plus grande clarté et précision dans leur formulation afin de rappeler l'existence et la particularité des droits moraux. Selon nous, une renonciation partielle aux droits moraux, dans le respect de la loi sur le droit d'auteur, pourrait être également envisagée et serait de nature à respecter la logique du logiciel libre.

Enfin, les licences de logiciels libres doivent, comme tous les contrats, respecter les exigences de formes prescrites par la loi sur le droit d'auteur. Ainsi, le caractère lacunaire ou imprécis des licences est théoriquement susceptible de les affecter de nullité en tout ou partie. En principe, ces conditions de forme se veulent protectrices de l'auteur dans ses relations contractuelles. Or, dans le contexte du logiciel libre, ce formalisme semble paradoxalement devenir un obstacle au développeur qui souhaite céder ses droits de manière la plus étendue possible.

Prendre le droit d'auteur comme base légale pour les licences de logiciels libres implique néanmoins de tenir compte des contraintes particulières posées par ce régime légal. Par conséquent, le choix de bénéficier des modalités offertes par le droit d'auteur doit s'accompagner du respect des spécificités des droits moraux et des exigences légales en matière de cession des droits. A défaut, le modèle du libre risque de rester du domaine de l'utopie, juridique à tout le moins.